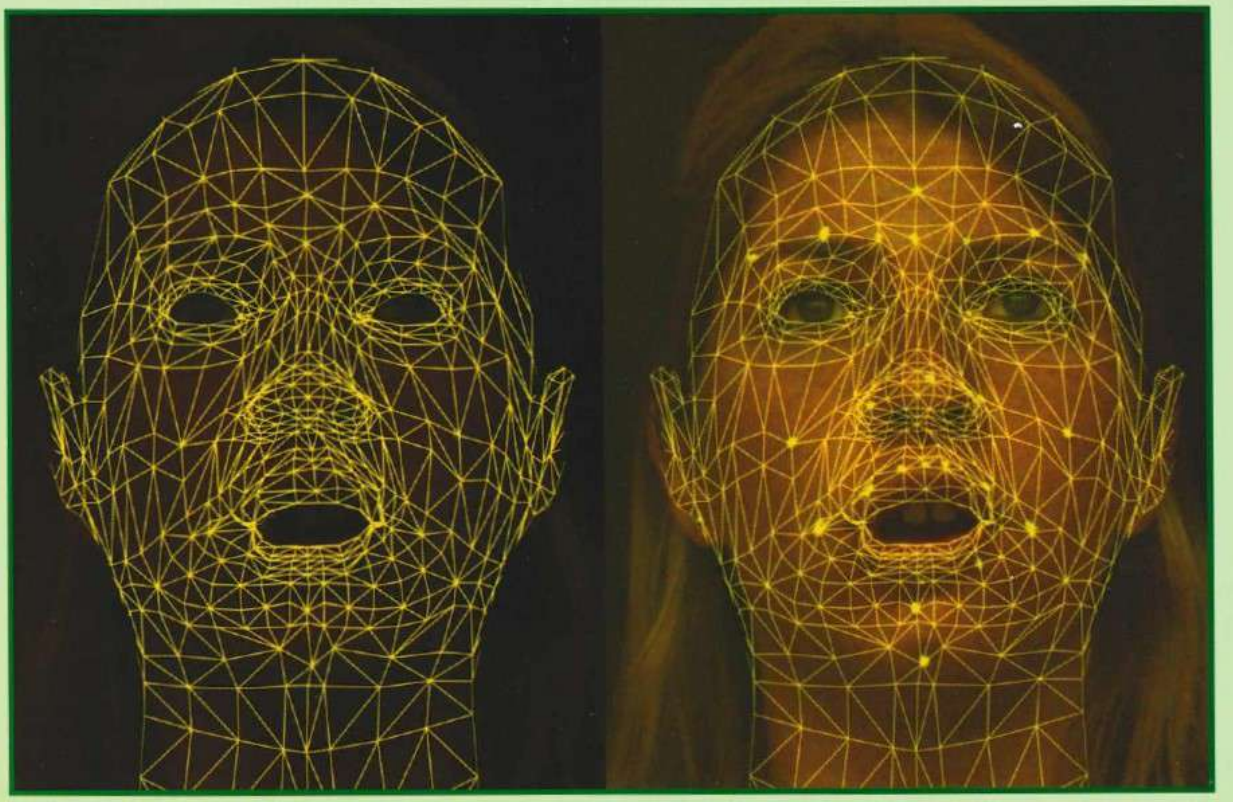
híradástechnika

1945 VOLUME LX. 2005

info-communications-technology



Speech research

Software development results

Filtering

Selected Papers **2005/6**

Journal of the Scientific Association for Infocommunications with co-operation with the National Council of Hungary for Information and Communications Technology

Contents

Cover: Speech research can be supported by facial modelling (a facial model works at the Pázmány University).

Foreword

How can quality be measured

lajtha.gyorgy@t-com.hu

In our English version we tried to collect the best papers of the last half year period. The aim is clear but it is difficult to define the specification of the "best papers". It is an often used phrase in several journals and conferences. In general it reflects a long discussion and a voting process. In spite of this the groups of experts participating in this process are not fully happy with the result. Namely the extremely new statements, or the high level content or the excellent style can influence the readers but the weighting of them is subjective.

The problem of selection in our English issue is based on special aspects. We consider that the readers of the English version are well educated, experienced professionals, so real novelties must be presented. Reviewing the famous periodicals published in English language we evaluated the up-to-date questions of the infocom field. We tried to fit our English version to the problems which are interesting all over the world. If we find no similar paper then there are to possibilities, either it is really new or it is out of date. We hope the first will be valid.

In accordance with this principle the first group of papers deals with sound and speech research. The sound of music instruments has a great importance solving music recording problems and in the course of planning music halls. This topic is quite seldom published in telecom journals, but this result has an impact on telecom development too.

The second group deals with the protocols used in telecom network engineering. The importance of software engineering is growing. The next generation of telecom network is based mainly of new software solution.

The filtering is a traditional technology in electronic engineering, but its content is enhancing. Nowadays new problems are arising. The spam and virus filtering is an important task.

Further on we have some interesting papers discussing ratification and telecommunication regulation. I try to collect this miscellaneous problem und the keyworld: Telecom Policy.

I hope at least one of these four broad general titles will be interesting for our kind readers. They are covering new interesting solutions in the field of telecom-infotech development the solution are offering more comfortable and rather safe methods. It means they may enhance the quality of the network, which is our primary goal.

In the next month Prof. Csaba Szabó is taking over my responsibility editing this periodical. I am sure he will take over all advantages of the last five years. He is an experienced and talented engineer who will modernize our methods improving the content. I hope you will be happy with the new structure of the journal and the new topics achieving higher priority in the selection.

Virtual speaker

LÁSZLÓ CZAP
Department of Automation, University of Miskolc
czap@mazsola.uni-miskolc.hu

JÁNOS MÁTYÁS
North Hungarian Regional Training Centre, Miskolc
matyasj@mail.erak.hu

Reviewed

Key words: facial animation, talking head, dynamic speech features, speechreading

Facial animation has progressed significantly over the past few years and a variety of algorithms and techniques now make it possible to create highly realistic characters. Based on the author's speechreading study and the development of 3D modelling, a Hungarian talking head has been created. Our general approach is to use both static and dynamic observations of natural speech to guide facial modelling. The evaluation of Hungarian consonants and vowels is presented for classifying visemes - the smallest perceptible visual units of the articulation process. A three level dominance model has been introduced that takes coarticulation into account. Each articulatory feature has been grouped to dominant, flexible or uncertain classes. The analysis of the standard deviation and the trajectory of the features served the evaluation process. Acoustic speech and articulation are linked with each other by a synchronising process. A filtering and smoothing algorithm has been developed for the adaptation either to the tempo of the synthesized or natural speech.

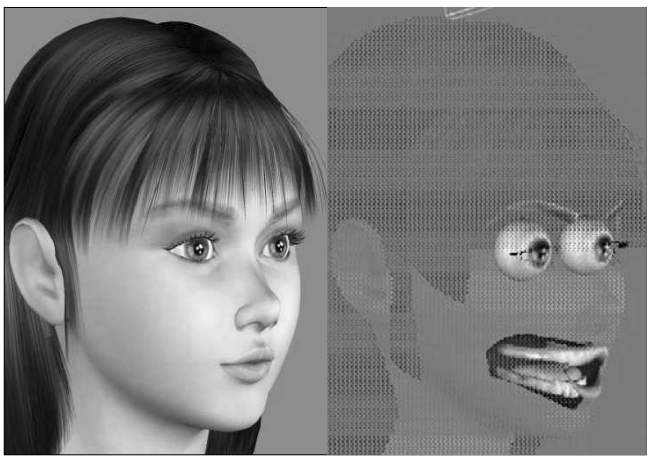
1. Introduction

The intelligibility of speech can be improved by showing the articulation of the speaker. This visual support is essential in noisy environment and for hearing impaired people. An artificial talking head can be a natural supplement to the sophisticated acoustic speech synthesis. The pioneer work of face animation for modelling the articulation started about two decades ago. The development of 3D body modelling, the evolution of computers and the advances at the analysis of human utterance enabled the development of realistic models.

Since the last decade the area has been developing dynamically and more and more applications have appeared. The audio-visual speech recognition and synthesis can open up a new prospect in the human-machine interface.

Virtual speakers and actors can improve the freedom of artists in multimedia applications. Teaching hearing impaired people to speak can be aided by an accurately articulating virtual speaker, which can make its face transparent and show the details of show the utterance better than a human speaker.

Figure 1.
Photorealistic and transparent visualization

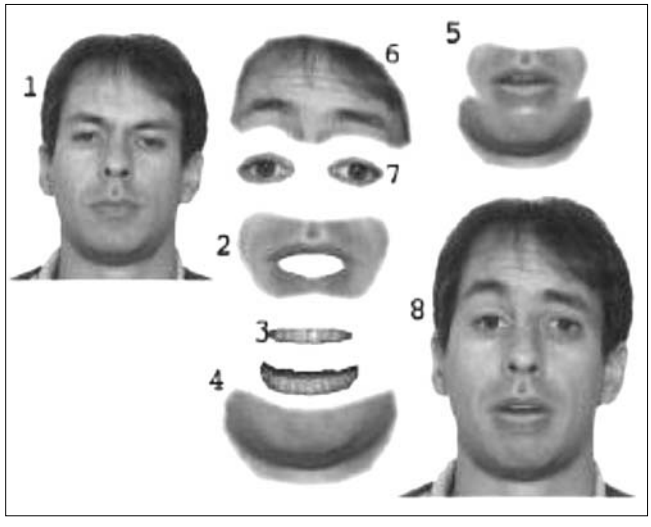


2. Speech animation

The first visual speech synthesizers were based on a 2D head model, recalling beforehand stored images of a speaker. Phases between keyframes sometimes were produced by image morphing. A 2D model can hardly provide head movements, gestures and emotions.

The progress at solid modelling directed the researchers' interest to the three-dimensional modelling. Either type of 3D models simulates facial expressions by tensing muscles. They produce realistic results, but the analysis of real muscular tensions is difficult. Surface

Figure 2.
Elements of a 2D head model [1]



models seem to be promising by acting textured polygons. Their features can be analysed on human speakers.

2.1. The visual unit of speech

The visual parallel of shortest acoustic unit, a phoneme is called *viseme*. The set of visemes has fewer elements than that of phonemes as utterances of several phonemes are visually the same. E.g. the voiced quality is invisible and the voices of the same place of articulation that are different only in duration or intensity belong to the same viseme class. The static positions of the speech organ for Hungarian phonemes can be found in essential publications. *Figure 3* shows the similarity of the same viseme on the speaker's photograph [5] and the 3D model [6].

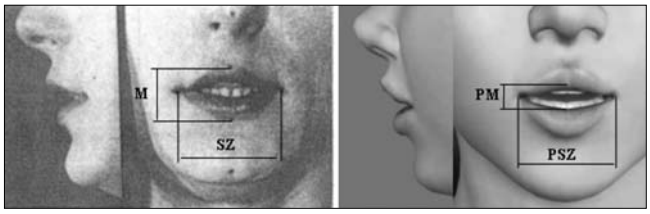


Figure 3.
A photograph of a speaker and the 3D model of the same viseme.

The features of Hungarian visemes have been created according to the word specimens of [4]. *Table 1.* shows the resultant groups of visemes represented by their Sampa codes.

Vowels	Consonants
E	b, p, m
e:	f, v
ii	t, d, n
2, o	r
y, u	s, z, ts, dz
A:	l
O	S, Z, tS, dZ
	t', d', j, J
	k, g
	h

Table 1. The Hungarian viseme classes

Remarks to the grouping:

- Viseme classes are based on the lip shapes, the invisible tongue position can be different e.g. o – 2, u – y.
- The lip opening of unlisted vowels are narrower than that of their short counterparts.
- For synchronization an enlarged selection is used.

The main features of visemes can be adapted from the published sound maps [4] and albums [5,6]. These features are the foundation of keyframes that the articulation is based on [7].

Features controlling the lips and tongue are crucial. Basic lip properties are the opening and width, their rate is related to lip round. The lip opening and the visibility of teeth are referred to the jaw movement. The tongue is described by its horizontal and vertical position, its bend and the shape of the tongue tip *(Figure 4.)*.

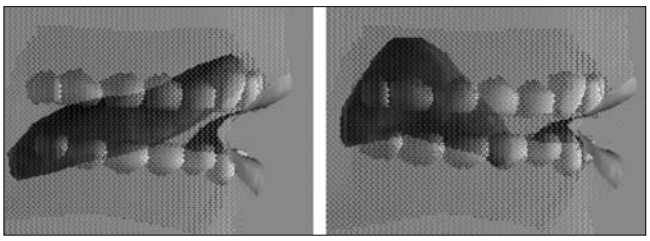


Figure 4.
Illustrative tongue positions for sounds n *(left) and* k-g *(right).*

Based upon the static features, the articulation parameters characteristic to the stationary section can be set.

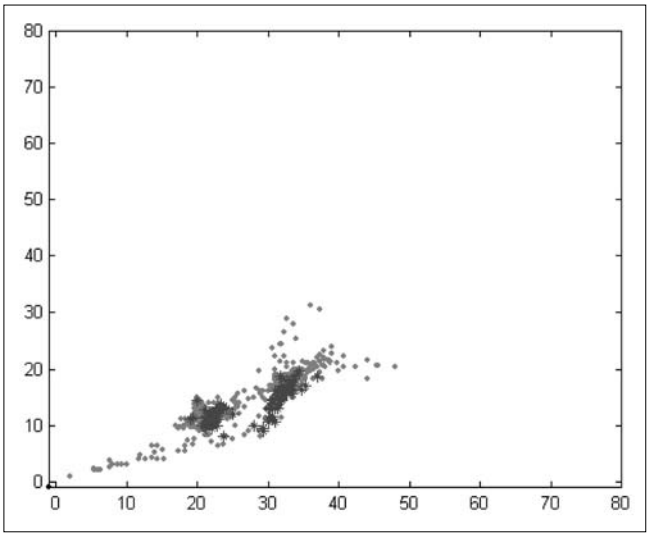
2.2. Dynamic operation

The dynamic features of continuous Hungarian utterances have not been described yet. The usage of motion phases represented in voice albums are limited, and can be related only to the particular word of specimen. The other source of dynamic analysis are my own studies in speechreading [8]. Trajectories of width and height of lips and the visibility of teeth and tongue are derived from there. These data drive the interpolation between the motion phases.

Some features take their characteristic value, while others do not reach their nominal value during utterance. All features of the visemes (eg. lip shape, tongue position) were classified according to their dominance. The categorization is based on the standard deviation of the speechreading data. Viseme features can be divided into three grades:

- *dominant* – coarticulation has no effect on them
- *flexible* – the neighbouring visemes affect them
- *uncertain* – the neighbourhood determines the feature.

Figure 5.
Lip open and lip width of transitional (.) and stationary (*) phases of S.



Besides the standard deviation, the distribution of transitional and stationary periods of visible features help to determine the grade of dominance. In *Figure 5.* the lip sizes of transitional and quasi stationary phases of sound *S* can be seen. Among the transitional states, determined by the neighbouring sounds, the features of the middle frames cover a restricted area.

The trajectory of viseme features can be also essential for determining the dominance classes. *Figure 6/a.* shows the trajectory of lip sizes of viseme *E*. Nevertheless, the curves cannot be traced one by one, but it is observable that they go through a dense area regardless of the starting and final states. The dominant nature of vowels' lip shape is obvious.

In contrast, the uncertain features do not tend to a certain value. The trajectory of *h* can be seen in *Figure 6/b.* (To be able to track them, only a couple of curves are represented.)

Figure 6.
Trajectory of lip sizes of viseme E *(a) and* h *(b)*

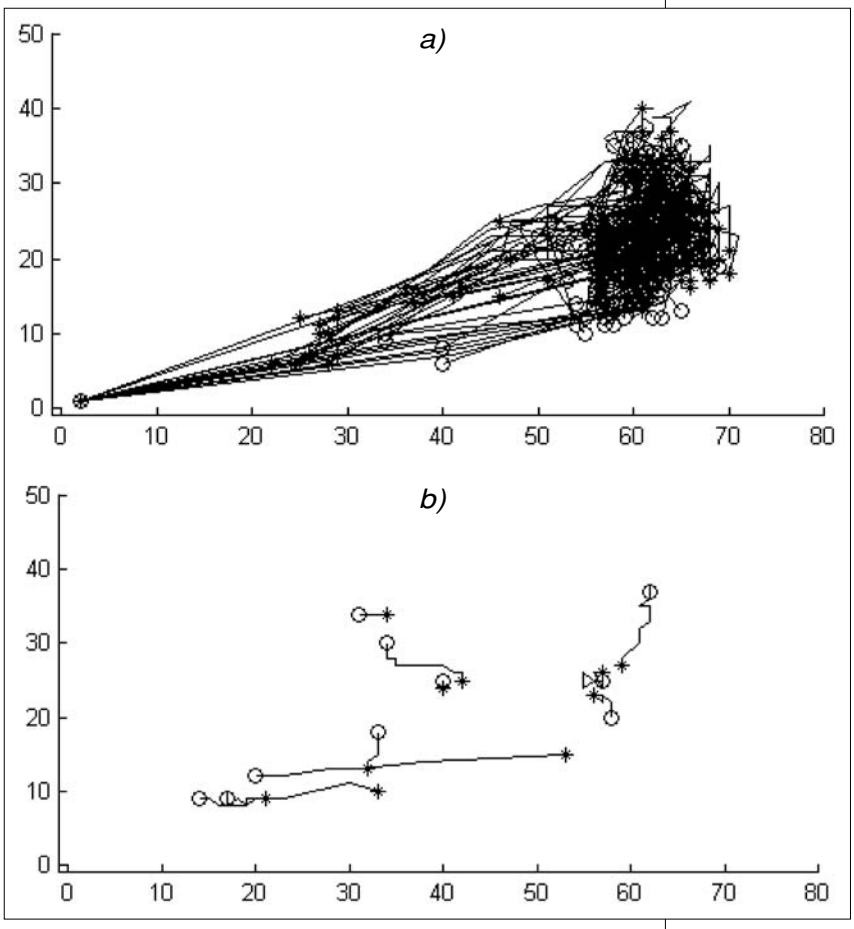


Table 2. describes the dominance classes of lip shapes while *Table 3.* shows that of horizontal position of tongue.

Table 2. Dominance grades of lip shape

Dominant	vowels, S, Z, tS, dZ
Uncertain	k, g, r, h
Mixed	p, b, m, l, j, n, J, f, v, s, z, ts, dz,, d, t, t', d' (lip opening is dominant, lip width is uncertain)

Dominant	t, d, n, r, l, t', d', j, J, S, Z, tS, dZ, s, z, ts, dz
Flexible	vowels
Uncertain	p, b, m, f, v, k, g, h

Table 3.
Dominance grades of horizontal position of tongue

The dominance grades of visemes control the interpolation of the features. Other improvements – as inserting a permanent phase into long vowels – refine the articulation.

3. Improving the naturalness

Studying the head movements of professional speakers, moderate nodding, tilting and blinking were introduced in a semi-random manner. Algorithm for head movement and mimicry can hardly be created according to prosody – these features are manually set by tags (e.g. lifting eyebrows at sentence accent, or control the glance). In dialog systems gestures can support the turn taking, the lift of eyebrows can indicate paying attention, nodding can mean acknowledgement.

3.1. Pre-articulation and filtering

Prior to utterance there is an apt. 300 ms silence period inserted – imitating breathing by opening the mouth – then the first dominant viseme is progressed from the neutral starting position. By this pre-articulation the mouth is formed before the sound is emitted in like manner as natural speech.

During the synchronization to natural or synthesized speech we were faced with different tempo of speech. When the speech is slow viseme features approach their nominal value, while fast speech is articulated roughly. For flexible features the round off is stronger in fast speech. A median filter is applied to interpolation of flexible features: the values of neighbouring frames are sorted and the median is chosen. A feature is formed by the following steps:

- linear interpolation among values of dominant and flexible features neglecting the uncertain ones,
- in the neighbouring of flexible features median filtering is performed,
- these values are then filtered by the weighted sum of the two previous frames, the actual and the next one.

The weights of the filter are fixed, not depending on the speech tempo. The smoothing filter refines the movements and reduces the peaks for fast speech.

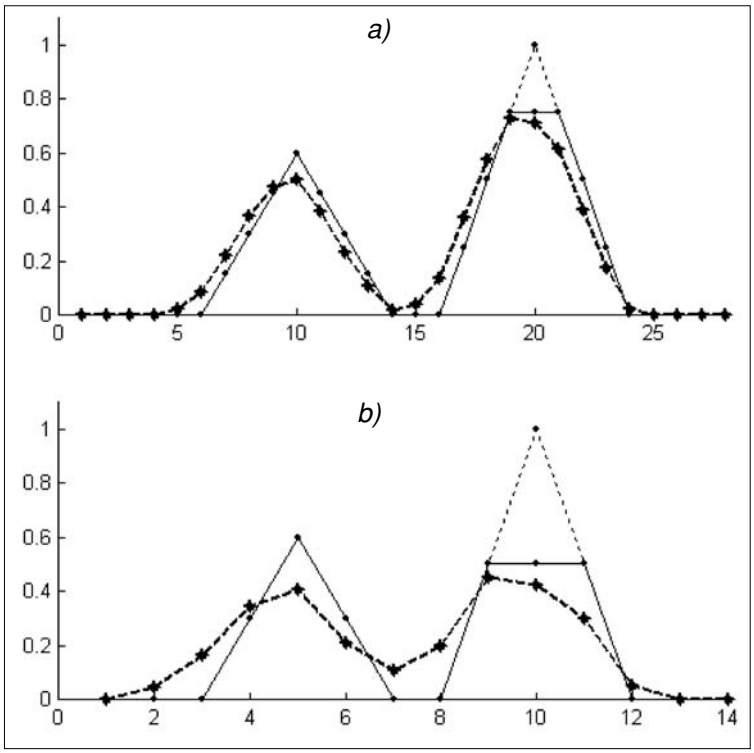


Figure 7. The interpolation of dominant (first peak) and flexible (second peak) features for slow (a) and fast (b) speech after linear interpolation (...), median filtering (—) and smoothing (---).

Figure 7. depicts the effect of median filtering and smoothing. In the example the slow speech contains double as many frames as the fast one.

3.2. Expressing of emotions

In multimodal speech we can confirm or disprove the verbal message by gestures and body language. After Ekman, the basic emotions can be selected in a scalable manner: anger, disgust, fear, enjoyment, sadness, surprise. *Figure 8.* depicts a couple of examples.

Figure 8. Expression of disgust and enjoyment

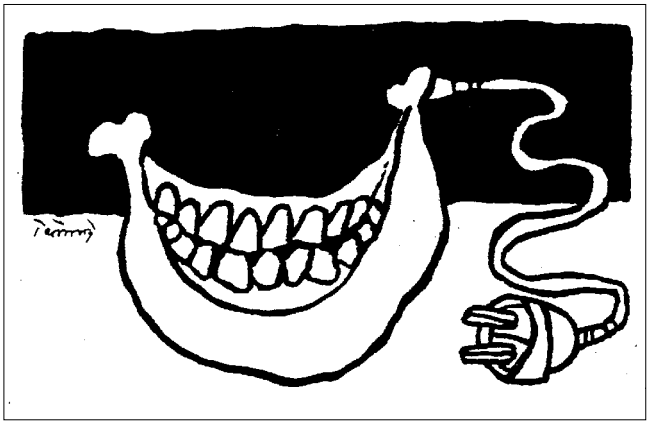
4. Conclusions

This paper describes the results of several years of research and development work that aim at working out a Hungarian audio-visual text-to-speech system. In this phase further refinement of co-articulation is performed. Due to the time consuming rendering process the virtual speaker can be utilized for reading pre-recorded messages. In the near future the results are going to be transferred to a real-time rendering platform.

Sample videos can be found:
http://mazsola.iit.uni-miskolc.hu/~czap/mintak

References

[1] Cosatto E., Grafat H. P. (1998)
2D Photo-realistic Talking Head.
Computer Animation,
Philadelphia, Pennsylvania, pp.103–110.
[2] Massaro, D.W.: (1998)
Perceiving Talking Faces.
The MIT Press Cambridge,
Massachusetts, London,
England, pp.359–390.
[3] Bernstein, L.E., Auer, E.T.: (1996)
Word Recognition in Speechreading.
Speechreading by Humans and Machines.
Springer-Verlag, Berlin Heidelberg,
Germany, pp.17–26.
[4] Molnár J.: (1986)
The Map of Hungarian Sounds.
Tankönyvkiadó, Budapest.
[5] Bolla K.: (1995)
A Phonetic Conspectus of Hungarian.
Tankönyvkiadó., Budapest.
[6] Bolla K.: (1980)
Hungarian Sound Album.
MTA Nyelvtudományi Intézet, Budapest.
[7] Mátyás J.: (2003)
Visual speech synthesis. M.Sc. Thesis,
University of Miskolc.
[8] Czap, L.: (2000)
Lip Representation by Image Ellipse. ICSLP 2000
Bejging, China, Proceedings Vol. IV., pp.93–96.
[9] Ekman, P., Friesen, W.: (1978)
Facial Action Coding System.
Consulting Psychologists Press. Inc.



Design issues of a corpus-based speech synthesizer

ANDRÁS NAGY, PÉTER PESTI, GÉZA NÉMETH, TAMÁS BŐHM

Budapest University of Technology and Economics,
Department of Telecommunications and Media Informatics

{nagy.a, pesti}@alpha.tmit.bme.hu

Reviewed

Key words: synthesized speech, speech quality, sampling, corpus volume

The corpus-based approach is a new technique which has never been used in Hungary. It offers a more flexible and better quality synthesis. This article outlines the basic principles of this technique then a more detailed description follows of the development of a Hungarian corpus-based, object-related system being under development at the Speech Research Laboratory of the Budapest University of Technology and Economics. In the second part of the article statistical studies with weather forecasts are introduced then some considerations regarding the selection of announcers are presented. Finally some other design issues of corpus-based systems are addressed.

1. Introduction

As a result of the convergence and integration of telecommunications, information and media technologies today, our world is headed towards the realization of an information based society. The process with the most profound effects in this transition – in addition to the advancement of networks, mobility and computers – is the transformation of human-computer interaction. Speech technologies, such as speech synthesis, play a fundamental role in this change.

Recent years have brought the new concept of corpus based speech synthesis [1]. The core idea of the concept is the generally accepted principle that the quality of a waveform concatenation based speech synthesizer will be determined largely by the number of concatenation points. As the length of the elements used in the synthesized speech increases, the number of concatenation points decreases, resulting in higher perceived quality.

In the ideal case, all possible texts, or at least all possible sentences would be stored as a single waveform element in the system's database. Since this is infeasible in a practical implementation, shorter units are introduced into the database, while aspiring to concatenate the output speech using few elements with high probability. The flexibility of the system makes it desirable to use variable length elements instead of a certain fixed length [2].

A number of speech synthesizers have been created for some languages spoken by many people following the ideas outlined above [2], but a Hungarian implementation was not yet available. The objective of our work is therefore to create such a modern speech synthesis system, building on results and experiences of previous Hungarian solutions (Profivox [3,4] and number synthesizer [5]). Since the creation of the required complex software is a task for several years, we are developing a closed domain synthesizer first for weather reports. This system allows simplified design, while serving as the foundation for a wide – or possibly unrestricted – domain synthesizer.

This article reports on the first phase of the research and development process. We give an overview of the primary challenges of corpus based synthesis, propose solutions to some of the problems, and present our initial experiments, which we use to evaluate the potential of the concept. The article is concluded by providing a summary of our work so far, and outlining the research and development tasks lying ahead.

2. Design challenges in corpus based systems

In this section, we give a brief insight into the challenges of designing corpus based systems, describe our experiments and possible solutions to some of the problems.

2.1. Speaker selection

Matching of waveform elements, originally cut from different parts of the speech corpus, is of paramount importance to the quality of the synthesized speech. This is ensured by an item selection algorithm, but the speaker's ability to produce speech with constant prosody properties predetermines the selection possibilities.

It is a basic requirement that the fundamental frequency (pitch) of the speaker should not fluctuate in a wide range, for example. Although the pitch can later be modified by signal processing methods, this intervention has an adverse effect on the quality of the synthesis. These considerations have led us to define some requirements to be met by a speaker.

These requirements were the following: clear articulation, pleasant tone, consistency (ability of the speaker to produce the same phonemes in a similar fas-

hion within one session, and also between sessions), and availability (sufficient amount of audio recordings accessible from the speaker). The eventual selection of a speaker was made in several steps based on the requirements.

We downloaded two full days of the broadcast archives for the stations available on the homepage of the Hungarian Radio (Kossuth, Bartók and Petőfi stations). The audio files were accessible in an hourly breakdown in RealAudio format, but the quality of these files didn't permit detailed acoustic analyses.

Characteristics of the speakers were collected by listening to the audio files multiple times. Comparison of these features with each other and with the initial requirements has resulted in our list of speakers deemed most appropriate for inclusion in the speech corpus.

We requested high quality audio recordings for the selected speakers from the internal archives of the Hungarian Radio. These files allowed more detailed investigations of acoustic properties, with the pitch and intensity as the most important aspects. We studied the values of these features on the time scale, the averages of the values, and the deviation from the average. The analysis was concluded by proposing the speaker with the most advantageous characteristics.

2.2. Issues of element selection

The key idea of corpus based synthesis is the availability of multiple versions of elements for concatenation during synthesis, making the selection of the best element possible according to a given metric. While in a diphone synthesizer the only consideration is the match of the phonetic labels of concatenated diphones, in the corpus based solution multiple aspects can be balanced with the use of a compound cost function.

The metric describing the correspondence between a selected element and the portion of speech to be synthesized is called the *target cost* [1]. The naturalness of the synthesized speech is strongly influenced by the match between elements concatenated together. This is captured by the *concatenation cost.* By definition, the concatenation cost of two neighboring elements from the speech corpus is zero, as the cut speech can be restored in its original natural form.

To investigate correspondence and matching, features are specified at the levels of phoneme, syllable, word and prosodic unit (such as a clause). Acoustic features of speech (such as the pitch and formant structure) are not currently utilized in our system, as we suppose that the prosodic features (such as the tone and modality of the sentence) hold sufficiently strong discriminative power. After tuning the weights of the cost function factors, the annotated speech corpus allows determining both the correspondence between a portion of speech and any part of the speech corpus, and identifying the fit between any two elements selected for concatenation.

Cost function factor weights can be adjusted by going through multiple iterations of listening test and modification phases. The correspondence of phonemes is not an absolute requirement, which has the important implication that phonemes of the same class can substitute one another, assuming that the concatenation cost is significantly decreased by this exchange. The utility of such a solution is explained by the fact that the imprecise phoneme may go entirely unnoticed by the listener if it fits well in the auditory environment (for example in an unaccented case).

Element selection cannot be done one by one because the fit of elements to each other must be taken into account. The goal of maximizing the overall quality of the produced speech makes a method similar to the Viterbi-algorithm [6] a plausible choice. Acoustic and prosodic effects overarching sentence boundaries can be disregarded, and therefore the target of optimal synthesis is a single sentence. The cost to be minimized is the sum of the target and concatenation costs for the entire sentence, over all possible selections of units.

2.3. Specifying element size

The peculiarity of corpus based synthesis is that in addition to making an element choice decision, the length of the element to be inserted can also vary [6,7]. When – in accordance with the requirements outlined in the previous section – the concatenation cost is zero for two adjacent elements from the speech corpus (the elements occurred together in the recording), then minimizing the cost function implicitly determines the size of the element as well.

However, this approach is not applicable in a real setting. The speech synthesizer is designed for a limited, but not closed domain, which means that knowing the target domain doesn't exclude the occurrence of new words (such as region names). To allow the synthesis of any arbitrary word, the system must be able to create speech from basic building blocks; diphone or triphone based synthesis must be available. The building blocks are necessarily the basic elements (diphones or triphones), if element size is specified with the help of a cost function. Search space can contain several million elements in this case, resulting in a long time to find the appropriate element – and eventually resulting in slow synthesis.

A possible solution is the *acoustic clustering* (AC, [8]) of elements, such that elements clustered together have minimal distances given by target cost function. Clustering can be done offline when annotating the speech corpus. The clusters can be used to reduce search space during synthesis. This approach has the advantage of not explicitly binding clusters to certain features.

In a different approach, longer elements (such as phrases, words or syllables) are also labeled in the speech database, and can be selected directly (without the implicit selection mechanism of a cost function)

[9,10]. The *Phonological Structure Matching* (PSM, [8]) algorithm first searches for an element to be inserted among the longer elements of higher levels. If this does not succeed, search is continued at a lower level. In the worst case (like when synthesizing a new word), the building blocks will be the diphone or triphone elements of the lowest level.

The PSM implemented in this manner still has to face the large number of different elements at the lowest level. This lead us to use acoustic clustering (AC) of diphones in our system below the segment level, while letting PSM select the element size above this level [8,11].

Consequently, at least one instance of all possible diphones must exist in the recorded speech corpus. To ensure this, we split the texts for the announcer to two parts, designed along different lines. The first part provides coverage of frequent words and phrases as determined from the statistical properties of the target domain, and allows selecting the longest possible elements for concatenation. The other part ensures coverage of diphones for the diphone based synthesis.

2.4. Database design and statistical analysis

The quality of corpus based synthesizers is fundamentally influenced by the construction of the speech corpus, which the element selection algorithm can later use to retrieve elements of varying sizes [8,12]. An efficient element selection algorithm assumes a well structured data storage solution. Care must also be taken to allow later potential extensions in the designed and realized database, without risking inconsistency.

The design of a well utilized speech corpus requires determining an optimal set of elements for storage in the database. Optimality in this case means finding an equilibrium between a large number of elements demanded by quality requirements, and a minimal element number constrained by performance considerations.

To help in finding an element set of optimal size and composition along these guidelines, we conducted some statistical analyses. Our investigations are based on a continuously growing collection of texts containing weather reports from various Hungarian sources on the Internet. The analysis database stores word forms and word form pairs, allowing statistical analyses for word forms, word form pairs and general statistical properties (such as the number and modality of sentences). A syllable-level analysis database is also under development.

The main table of the database contains word instances. Every word instance has an identifier and type (word, number, abbreviation, sign, punctuation), and the identifier of the preceding and following words, position in sentence, and sentence position in text are also stored. The position of the word in its sentence is recorded by two – numerical and structural – properties. The former means the number of the word in the sentence, while the latter shows whether the words is at the beginning or end of a sentence, preceding or following a comma, or in an enumeration. Any word instance can belong to more than one of these categories.

Before starting with the statistical analyses, we created a data table of abbreviations and their resolutions, storing the frequency of occurrence as well. A list of common misspelled words was also created, giving the correct form of the words and the frequency of the misspelled version. The construction of these tables was helped by a certain level of automation, but was mostly done by hand. In practice, abbreviation resolution was done by looking for features of words indicating an abbreviation, – such as three-letter long words containing only consonants are typically abbreviations, – and then reviewing the list manually. We used an external spell checking solution to collect misspelled words from our text corpus.

Our investigations revealed that the most common types of errors were mistakes in accentuation. The normalization of the text corpus was accomplished using the abbreviation and misspelled word tables. It is worthwhile to note that these tables will be of further use in automating the correction of new weather report texts.

The 20 sources of weather reports (such as *http://www.met.hu*) provided 56,000 sentences, containing 670,000 elements (words, numbers, abbreviations, signs and punctuations – signs are the “+”, the “-”, the “plus” and “minus” words) between April 2004 and May 2005. Some 493,000 of these are words (5200 distinct word forms), 43,000 are numbers, and the rest are punctuation and signs. Almost all sentences are statements; there were only a few questions and exclamations. On average, there are 10 words in a sentence (including numbers as well). The average length of words is slightly over 6 letters, which might seem unintuitive, as the list of most frequent words are topped by definite pronouns, one or two letters in length. The explanation lies in the frequent presence of longer than average weather related expressions (such as “hőmérséklet”, “várható”, “csúcsértéke”, “felhőzet”; *“temperature”, “expected”, “peak”, “clouds”*). The length of the longest word is 23 letters (“hőmérséklet-csökkenéssel”; “with decrease in temperature”). Hyphenated words were regarded as a single word (such as “Dél-Dunántúl”; *“South-Transdanubia”*).

A table of word length distribution was created, which showed that words between lengths of 6 and 10 appear in the most various forms. These, and further investigations of words include words in the traditional sense only, and do not include numbers and punctuation.

The frequency of word forms was also investigated: our list of words was labeled with the coverage percentage provided by each word. For the k^{th} word this means that a list of the k most frequent word forms would cover a portion of our weather report corpus; the size of this portion is given by the sum of coverage percentages for the k words.

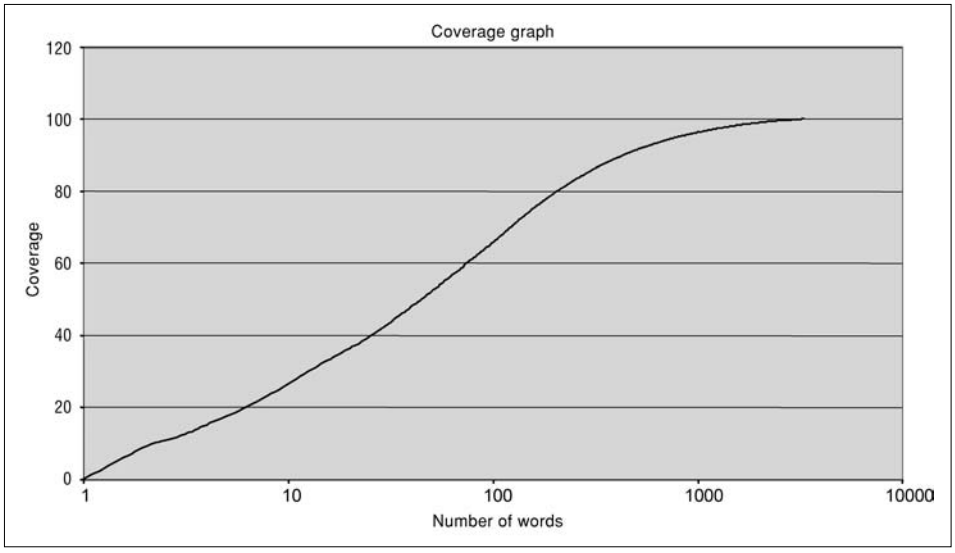


Figure 1. Coverage graph for weather reports

Our analysis led to the conclusion that the 10 most frequent word forms cover 31% of the input corpus. As little as 500 words ensure 92% coverage, while with 2300 words this reaches 99%. A corpus from an unrestricted domain requires approximately 70,000 word forms to reach 90% coverage [13].

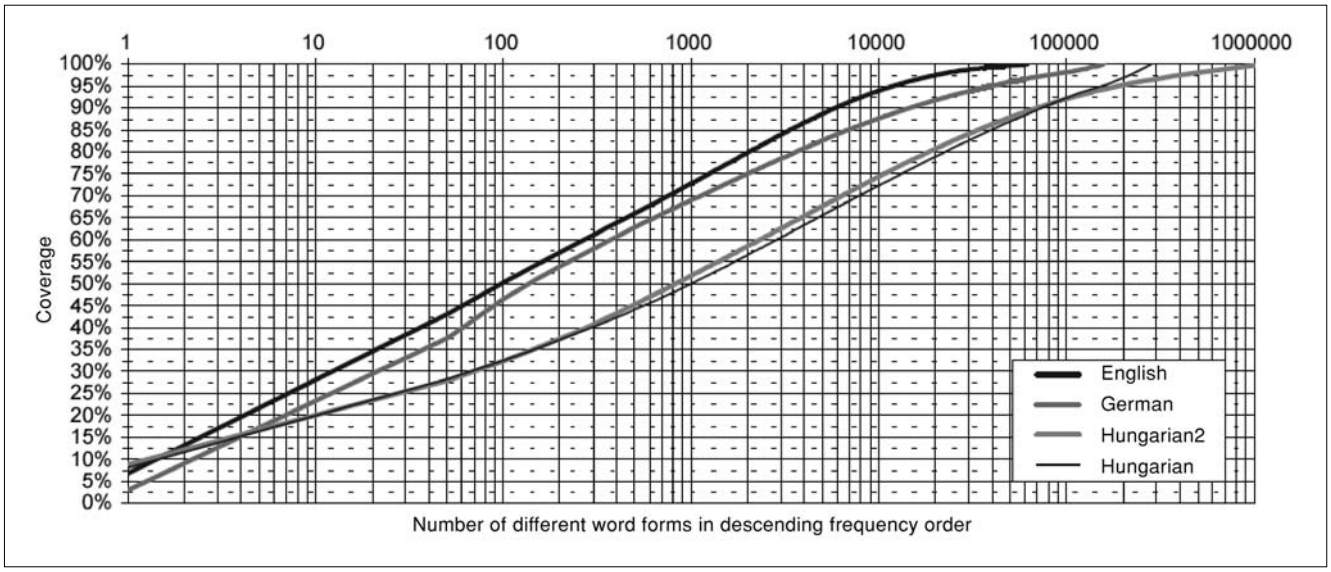
This favorable property is the result of the restricted domain, and in part a result of the limited annual coverage of the weather report corpus (the unavailability of archived weather report texts limited us to one year of our data collection period). Our further inquiries will concentrate on factoring out the latter. However it is important to notice, that these properties are almost exactly the same as the ones we got from our previous analysis based on a half year collection period, so we assume that these results are mainly due to the restricted domain. The coverage diagrams for the restricted and general domains are shown on *Figure 1.* and *Figure 2.* for reference.

Any word form must be available in multiple surroundings, as the pronunciation varies with the word's position in the sentence or with the different adjacent sounds. Considering this need, the final corpus will contain more word forms than the minimum required for simple coverage. This has led us to create statistics concentrating on word form frequencies in view of the left and right context. Frequent words (such as "hőmérséklet"; *"temperature"*) need to be recorded occurring in most of their frequent context types.

In order to take into consideration the position of a word inside the sentence, our data table was created to store information on this aspect as well. The actual position number is of little importance in a practical solution, since structural information captures location dependent word realization in sufficient detail. This structural information represents different pronunciation requirements as the beginning or end of a sentence, and before or after a comma (at clause boundaries and in enumerations). The design of the speech corpus has to incorporate this knowledge. In addition to investigating properties of individual words, the statistical analyses have to deal with words pairs (and word chains in the general case), to allow high quality synthesis of frequent phrases.

Finally, we created a list of foreign words appearing in weather reports (such as "Dubrovnik"), as these have to be synthesized differently. A possible solution is an exception dictionary, containing the correct Hungarian pronunciation of these foreign words ("Dubrovnik"

Figure 2. Coverage graphs for the general domain (source: [13])



would be translated to “dubrovnyik”, which corresponds to the Hungarian pronunciation of this word).

Considering the construction of the speech corpus, the design process has to select a small set from the available large text corpus, providing a good coverage for the entire corpus. A greedy selection algorithm is commonly used to achieve this goal [12]. This is a simple iterative solution, selecting entire sentences from the large text corpus, guided by a target of high coverage for the input corpus. Each iteration adds a sentence containing the highest number of words that are not yet covered by the selected corpus. An element is considered to be not covered if the corpus under construction does not contain an element with the same feature vector (a vector with parameters of interest (such as tone or intensity) as dimensions. Values along dimensions express how much a certain property is valid for the element. The iterative process terminates when the corpus under construction fulfills some predefined requirements (such as providing a given coverage ratio).

A key point to the success of the algorithm is the size and composition of the feature vector. A long vector results in high differentiation between elements, and results in the algorithm failing to cover most of the elements. A short vector will results in multiple coverage of most elements, prohibiting the selection of the most appropriate one. An optimal solution is not known, but several proposals outlining the composition of the feature space are available (such as emphasizing stress or the left and right contexts). A traditional solution associates a boolean value to features (criteria fulfilled or not), but fuzzy implementations, allowing a continuum of values between 0 and 1 also exist, such as in considering a match of left contexts. Stress is best represented with two possible values (stressed or unstressed). The success of the fuzzy solution mostly depends on the mapping of feature fulfillment levels to non-binary values.

The algorithm is generally used to obtain full diphone coverage, but can be extended to also select sentences that contain elements (words, word pairs) that are worthwhile for inclusion in the created corpus, based on their frequent occurrence shown by the statistical analyses. As mentioned previously, the investigations need to consider factors other than the frequency of occurrence, such as the context.

Database design must also consider the LNRE phenomenon (*Large Number of Rare Events,* [14]), which means that while the majority of elements rarely occur in speech, – and therefore each of these elements are rarely used in synthesis, – the number of such elements is so significant that at least one will be necessary for the synthesis of any given sentence.

Since the creation of the speech corpus focuses on including the most frequent syllables, words, phrases and sentences, the existence of the LNRE phenomenon implies that virtually all sentences requested for synthesis will contain portions that have no corresponding elements in the database. This means that the corpus must be constructed to include all possible diphones in at least one version, but the more frequent ones in multiple contexts. Such a design ensures that all portions of an input text can be synthesized to speech; using diphones in the worst case [11]. The number of possible diphones equals the square of the number of phonemes plus one (as silence can also be part of a diphone). However, not all diphones are necessary, as full diphone coverage can be achieved with at most a couple of thousand elements in European languages [15].

Once the speech corpus has been created along the above guidelines, it must be stored in a data structure allowing efficient element selection. This data structure is comprised of three fundamentally different parts.

The first is the collection of files containing the waveforms. Each file contains the annotated waveform of a single sentence. This solution, while ensuring a small file size, also allows using one file to load an element, since synthesis doesn’t require elements overlapping sentence boundaries.

The second part of the data structure contains the diphones. The directory of diphones is stored in a tree data structure, containing feature vectors and references to the files containing the diphones. The tree contains elements in the order of their appearance in the corpus, such that an inorder traversal of the tree returns the original corpus [16]. This is relevant in allowing a simple implementation of the varying element size selection. To select an element longer than a single diphone, the diphones returned along the inorder traversal of the tree initiated at the starting element can simply be concatenated.

The third part of our data structure speeds up search in the tree. We created a word tree [17], storing diphones such that nodes of this tree correspond to possible prefixes of the diphones, and the leaves of the tree contain references to diphones in the first tree.

When retrieving an element starting with a certain diphone, the leaf referencing this diphone is first located in the word tree (multiple matches are stored in a chained list). The requested longer element can then be obtained by the inorder walk in the corpus mapping tree.

The hierarchy described above can be improved by creating word trees for words, word pairs and sentences, in addition to making one for diphones. Creation of the data structure must ensure easy maintainability of consistency. As an extension of the database would take place before synthesis, allowing an update of the related search structures in addition to the update of the waveform files.

2.5. Listening experiments

We performed a proof of concept experiment based on the weather forecasts collected from the online archives of the Hungarian Radio. The weather forecasts of two consecutive days on radio stations Kossuth, Pe-

tőfi and Bartók were used. Although the linguistic content of these utterances were given, we could use them to evaluate the capabilities of the system being developed. Our aim was to predict the naturalness of the speech produced by a corpus-based system by manually synthesizing sentences of weather forecasts.

We analysed 149 weather forecasts by 22 announcers. There were just a small number of recordings available from each announcer that made the synthesis task harder than that of the system under development. Several words occur only once for some speakers. On the other hand, a number of words occur in almost every forecast since the recordings are taken from a two day period. During this short period there was not enough time for the forecasts to change much.

In order to get comparable results, we hand synthesized sentences on some announcers' voice that are also available on another voice. This is the rationale behind not choosing arbitrary sentences to be synthesized. The small corpus size was another reason. First, we selected the announcers with the highest representation in the corpus. Then we transcribed the forecasts belonging to them. In the transcript of the resulting 54 weather forecasts, we picked five sentences that consist of words that are also available by another speaker in a similar context. The details of the selected sentences are summarized in *Table 1.* where a horizontal line in a sentence denotes a concatenation point.

The stimuli for our listening experiment consisted of the five hand synthesized sentences, five natural utterances and five-five sentences synthesized by the diphone-based and the triphone-based Profivox text-to-speech system with automatic prosody (the triphone-based system is under development).

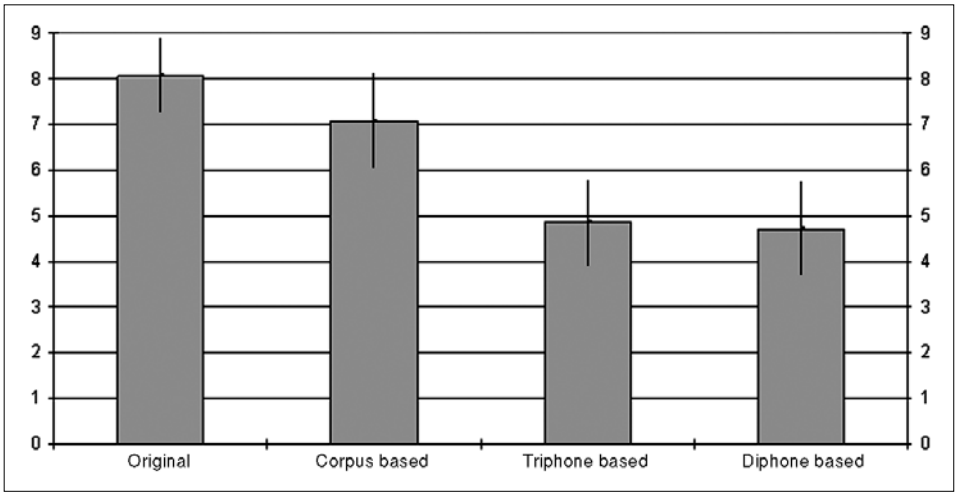


Figure 3.
Results of listening tests: average and standard deviation

The five native Hungarian subjects were instructed to rate the naturalness of each one of the 20 recordings on a scale between 1 and 9. The means and standard deviations of the ratings are shown on *Figure 3.*

According to the results, the triphone-based approach showed little improvement over the diphone-based system. Note that the former one is not a deployed system yet (among others, new signal processing algorithms are being implemented). The manually synthesized corpus-based sentences achieved a score two points higher than the systems employing concatenation of short, fixed-length units. The corpus-based stimuli was outperformed only by the natural utterances. The standard deviation is relatively small for each group that allows us to draw general conclusions. As it was expected, the scores of the natural utterances showed the lowest variability.

Table 1. Properties of synthesized sentences

Sentence	Original speaker	Speaker of synthesized sentence	# of elements concatenated
A hőmérséklet hajnalban mínusz | egy, | mínusz| hat, | holnap | napközben | mínusz | egy, | plusz négy fok | között alakul.	András	Adrienn	10
Napközben | országszerte várható csapadék, északon, | északnyugaton | havazás, | délkeleten eső, | másutt | havas | eső, | ónos | eső.	Erika	Klára	10
Végül az időjárásról: | mindenütt beborul | az ég, | reggelig egyre többfelé | lehet gyenge havazás, | hószállingózás.	Zsuzsa	András	6
A hőmérséklet | kora délután | kettő | és | hét | fok | között | alakul.	Zsuzsa	András	8
A nyugati, | északnyugati | szelet | sokfelé erős, | a Dunántúlon | helyenként | viharos | lökések | kísérik	Zsuzsa	István	9

Our experiment predicts a significant improvement in naturalness for the corpus-based text-to-speech system. Essentially, it is not expected that systems based on the concatenation of fixed-length units can achieve a comparable speech quality improvement. The corpus-based approach also has its drawbacks, such as a large database has to be recorded, labeled, stored and searched. Furthermore, the domain is limited and the computational complexity is far higher than for fixed-length concatenation.

3. Conclusion

The corpus-based approach to text-to-speech synthesis is a novel concept that has not been applied to Hungarian yet. It opens the way for synthetic speech to approach the quality of natural speech.

In this paper we outlined the fundamentals of this concept and gave a detailed progress report on the ongoing research at the Budapest University of Technology and Economics, TMIT Laboratory of Speech Technology (BME TMIT Beszédkutatási Laboratórium). The goal of this research project is to develop a limited domain, corpus-based TTS for Hungarian.

We gave an account of the statistical analysis performed on weather forecasts, the method of voice selection and discussed some other design issues. We conducted a listening experiment in order to predict the quality improvement achievable by the corpus-based approach.

Encouraged by the promising results, the next phase of our project is the implementation of the system. We are developing an algorithm for unit selection on multiple levels based on the results of our statistical analysis of the target domain. In the first version, we plan to define word and word N-gram levels and to restrict the input to a limited vocabulary. The second version would enable the synthesis of arbitrary words by backing off to a set of diphones based on acoustic clustering. The weights for the features used to calculate concatenation and target costs will be estimated by a sequence of iterative listening tests and refinements.

Acknowledgement

The authors would like to thank their colleagues at the BME TMIT Laboratory of Speech Technology for their invaluable help. We especially thank the Hungarian Radio for authorizing access to high quality weather report recordings.

References

[1] Bernd Möbius, “Corpus-Based Speech Synthesis: Methods and Challenges”, Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (Univ. Stuttgart), AIMS 6 (4), pp.87–116., 2000.

[2] Yi, J.R.W., Glass, J.R., “Natural-Sounding Speech Synthesis using Variable-Length Units”, Proc. ICSLP-98, Sydney Australia, Vol. 4, pp.1167–1170., 1998.

[3] Olaszy, G., Németh G., Olaszi, P., Kiss, G., Gordos, G., “PROFIVOX – A Hungarian Professional TTS System for Telecommunications Applications”, International Journal of Speech Technology, Vol. 3, Numbers 3/4, pp.201–216., Dec. 2000.

[4] Olaszi Péter, “Magyar nyelvű beszéd-szöveg átalakítás: nyelvi modellek, algoritmusok és megvalósításuk” (Hungarian Text-To-Speech Synthesis: Linguistic Models, Algorithms and their Implementation) PhD dissertation, BME, Budapest, pp.5–15., 2002.

[5] G. Olaszy, G. Németh, “IVR for Banking and Residential Telephone Subscribers Using Stored Messages Combined with a New Number-to-Speech Synthesis Method”, in D. Gardner-Bonneau ed., Human Factors and Interactive Voice Response Systems, Kluwer, pp.237–255., 1999.

[6] Jon Rong-Wei Yi, “Natural-Sounding Speech Synthesis Using Variable-Length Units”, Master of Engineering Thesis, Massachusetts Institute of Technology, 1997.

[7] S. P. Kishore and Alan W. Black, “Unit Size in Unit Selection Speech Synthesis”, Eurospeech 2003, pp.1317–1320., 2003.

[8] Antje Schweitzer, Norbert Braunschweiler, Tanja Klankert, Bernd Möbius, Bettina Sauberlich, “Restricted Unlimited Domain Synthesis”, Eurospeech 2003, pp.1321–1324., 2003.

[9] Eric Lewis and Mark Tatham, “Word and Syllable Concatenation in Text-to-Speech Synthesis”, Eurospeech 2001, Vol. 2, pp.615–618., 1999.

[10] Eric Lewis and Mark Tatham, “Automatic Segmentation of Recorded Speech into Syllables for Speech Synthesis”, Eurospeech 2001, pp.1703–1706., 2001.

[11] Michael Pucher, Friedrich Neubarth, Erhard Rank, Georg Niklfeld, Qi Guan, “Combining Non-uniform Unit Selection with Diphone Based Synthesis”, Eurospeech 2003, pp.1329–1332., 2003.

[12] Baris Bozkurt, Ozlem Ozturk, Thierry Dutoit, “Text Design for TTS Speech Corpus Building Using a Modified Greedy Selection”, Eurospeech 2003, pp.277–280., 2003.

[13] G. Németh, Cs. Zainkó, “Word Unit Based Multilingual Comparative Analysis of Text Corpora”, Eurospeech 2001, pp.2035–2038., 2001.

[14] Ove Andersen, Charles Hoequist, “Keeping Rare Events Rare”, Eurospeech 2003, Vol. 2., pp.1337–1340., 2003.

[15] Dr. Gordos Géza, Takács György, “Digitális beszédfeldolgozás” (Digital Speech Processing, in Hungarian), Műszaki Könyvkiadó, pp.191–197, 1983.

[16] Rónyai L., Ivanyos G., Szabó R., “Algoritmusok” (Algorithms, in Hungarian), Typotex, p.60., 1999.

[17] Knuth, D. E., “A számítógép-programozás művészete”, (The Art of Computer Programming, in Hungarian), Műszaki Könyvkiadó, Budapest, p.503., 1988.

New protocol concept for wireless MIDI connections via Bluetooth

CSABA HUSZTY, GÉZA BALÁZS

Budapest University of Technology and Economics, Dept. of Telecommunications and Media Informatics
info@midioverbt.com

Reviewed

Key words: wireless MIDI (Musical Instrument Digital Interface), Bluetooth

This paper describes a new protocol concept for wireless MIDI connections via Bluetooth. For the practical appliance, the protocol design supports nearly arbitrary connection topology. We show the plans of a generic Bluetooth-based MIDI system and describe the main ideas of its data transmission protocol, calculate its latency and investigate its limits of usability, while suggesting a few possible extensions to this system to be further realized.

1. The limits of the MIDI standard

The main function of MIDI is the synchronisation of the system beyond the transmission of control informations required for sound synthesis [1]. Devices utilizing MIDI protocol are equipped with 3 connector types. The IN connector is physically connected to the OUT connector of the adjacent unit, and the THRU output allows the user to form a chain topology of MIDI devices by sending the inbound data right to the device's THRU port. Using this method supports only connection topologies of limited structure without using auxiliary units.

It is also a quite common problem to combine more outputs into one input, which can only be solved with an external device, the MIDI Merger, too. Considering the maximal length of a few meters of the connection cables, the devices to be connected cannot be placed in an arbitrary order and distance, and further considerations have to be made if more than 15-20 devices are connected because of the increasing size, price and delay of the routing and switching units. We will not deal with the limits of the standard concerning the implementation henceforth.

2. Applying Bluetooth for MIDI connections

Among others Bluetooth is one of the inexpensive wireless solutions for the replacement of MIDI connections. The power consumption, the sufficient range and the prosperous noise resistance of the single units also make it applicable for this purpose. MIDI connections formed with Bluetooth can eliminate not only the cable but the other devices needed for the connection, so that they can be integrated into such existing devices. Although presented in a different way in other works [4], this paper describes a new method for realizing MIDI connections beyond the functionalities a MIDI cable offers.

Choosing the proper connection and the packet type for MIDI

The current 1.2 standard of Bluetooth supports one master and seven active slaves per piconet, although much more than seven slaves can be connected to it in parked state [2,3].

The master controls the channel access. All the other participants' clocks in the piconet are synchronized to the clock of the master and every unit is synchronized to the master in hopping frequency, too. The master may start a transmission only in even slots and a reception only in odd slots, while the slaves may do this vice versa: they may start a transmission only in even slots.

The above apply only for the initiation of the transmission: it can take up more than one slot, but the length must be an odd number of slots.

We choose the ACL type transmission for the MIDI application. One piconet supports one such type of data channel. The standard defines seven packet types for ACL connections, the maximum number of transmissible bytes are shown in *Table 1.*

We use the M-type packets for the data transfer, because they utilize 2/3 FEC encoding for error detection and correction, while the H packets feature neither error checking nor correction.

Table 1. Maximum data transfer rate in bytes of BT packets

packet type	max. tr.rate
DM1	18
DH1	28
DM3	123
DH3	185
DM5	226
DH5	341
AUX1	30

Hub-based topology and its advantages

The hub-based topology *(Figure 1.)* of the Bluetooth piconet facilitates the MIDI application. With broadcast type messages and correct settings any connection topology can be implemented, including the connections modifying the data stream, too.

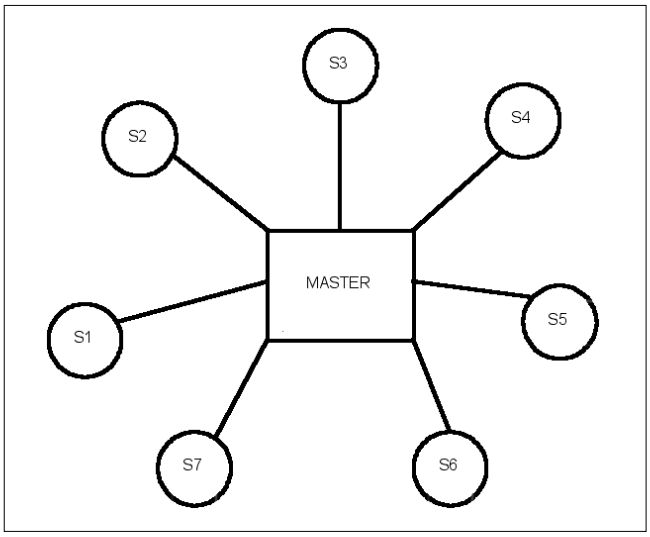


Figure 1. Hub-based topology

When setting up the connection we already know which slave unit will serve as a MIDI In and which as a MIDI Out. The Bluetooth based implementation includes the collateral possibility of looping back one's output to itself as an input – which is usually called 'MIDI Echo'.

Implementing basic MIDI connection types

To be able to create any type of connection topology, we have to implement the logical connections replacing the MIDI In, MIDI Out, the MIDI Thru Box (Hub), MIDI Merge, Echo and Patch Bay, either are they devices or functions.

1. MIDI Out/In (MIDI cable)

The data of the MIDI devices connected to the slave units get to the other device(s) indirectly via the master device. The user assigns the In and Out ports. The master device polls the first device, which sends its MIDI data back to the master. This data will be broadcast by the master to every active slave. Based on the set topology the S1 unit may ignore the incoming data (see *Figure 2*.).

2. MIDI Hub (Thru Box)

It is an easy task to connect one input to multiple outputs with these logical connections. After being polled by the master, the S1 slave sends its MIDI data. The only output port in this system is S1, so the polling ends after this point, and the master passes the data in a broadcast message to every slave. Since S1 is not configured as an input, it will discard the incoming MIDI messages. The constant delay of the arriving data with even more than one slave is a collateral advantage.

3. MIDI Merge Box

To unify two or more inputs in one output the master polls S1, which responds by sending its MIDI data. This data is temporarily stored in the master device for the broadcast packet, which is to be sent later, after polling S2 and getting its data. Slaves not configured as an input will discard the received data. There is need, however, to process the MIDI data streams before they are being merged for keeping its consistency, but the S3 could easily do this locally before sending the data to the MIDI device.

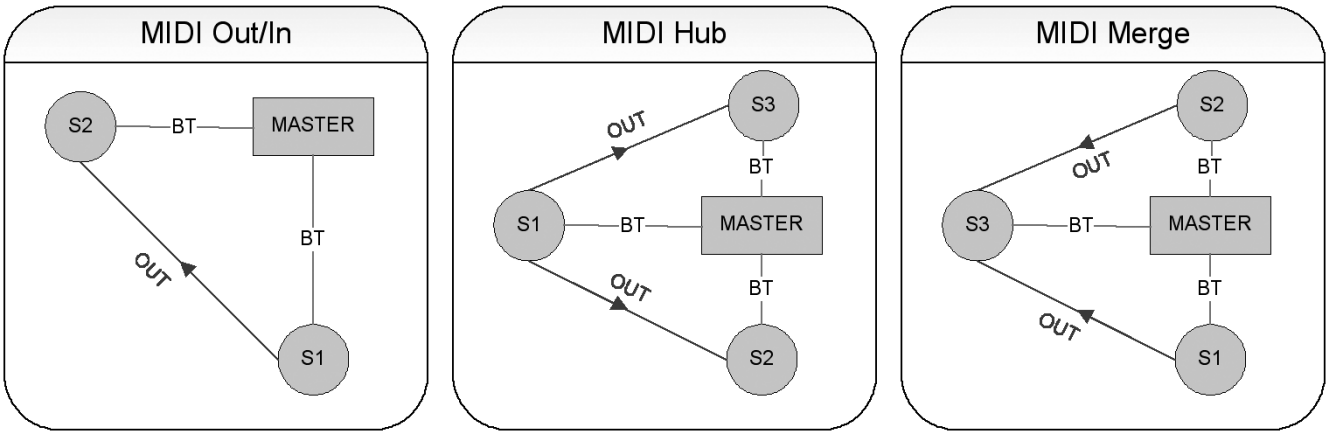
The suggested protocol concept and its timing

The logical units of the MIDI data, the messages have to be collected and split into packets when using the Bluetooth system. The MIDI bytes in the packets transmitted in a time unit (MIDI slot) are always broken on a message boundary. One exception exists: the System Exclusive message (SysEx), which can be of an arbitrary length. The incoming MIDI bytes are read one after the other from the MIDI Out ports by the predefined topology, which then form packets of constant length for each Out ports and are sent to the appropriate In-s. In some cases the packets have to be processed in order not to exceed the MIDI bandwidth (e.g. when merging). Such a cycle is called a MIDI slot from now on, which must have a firmly constant length in time. Considering the possibility of the reception of a variable number of bytes in a MIDI slot, the packet length and the delay time have to be computed for the case of the maximal byte count.

The timing of a MIDI slot (using DM3 packet with single transmission) is as follows:

1. The master polls the first slave that connects to a MIDI device with an Out port in the 0th Bluetooth (BT) slot [5].

Figure 2. Realizing MIDI connections



2. The addressed slave replies with a DM3 packet of constant length containing the received MIDI data. This transmission begins in the 1st and ends in the 3rd Bluetooth slot. If there are more than one Out-s in the system, the described process from step 1. is applied to each of them.
3. 2 empty BT slots follow, and then a DM3, DM5 or DH5 type of packet (depending on the number of the Out-s) is broadcast to every slave. The master device starts to construct the broadcast packet to be sent while still receiving data from the slaves (the UART ports of the BT modules and the Host Controllers are full-duplex), so there is only a little time to wait for the insertion of the data of the last slave. Since an even BT is coming after the data packet of the last slave, the system has to wait for the next even BT slot before it can send the broadcast message. This causes an additional delay of 2 BT slots.
4. Finally one empty BT slot comes, because the next polling sequence may start only in an even BT slot and the broadcast packet may not be responded. The whole cycle repeats from step 1.

To sum up, the amount of the needed BT slots are:

$$N_{BT\_SLOTS} = 4*O + 2 + x + 1$$

where O is the number of Out-s and x is 3 in the case of using DM3 and 5 when using DM5 packets for broadcasting.

Let B be the length of the MIDI slot in bytes:

$$B = S_{MIDI} \cdot T_{BT},$$

where S_{MIDI} = 3125 bytes/s, the transfer speed of the MIDI line and T_{BT} = 625 µs, the length of a BT slot.

In the interest of the planning of the timing let us calculate the maximum need of byte count to be transmitted. In the case of 1 active Out the maximum number of the transmitted MIDI bytes in a MIDI slot is:

$$B_{1\_OUT} = S_{MIDI} \cdot T_{BT} \cdot (1+1+2+1+1) =$$
$$= 6*S_{MIDI}T_{BT} = 11.71875 \Rightarrow 12 \text{ bytes}$$

where we rounded the sum up, considering the worst case. The terms of the sum are: 1 poll from master to slave, 1 response from slave to the master, 2 empty BT slots, 1 broadcast from the master to the slaves and 1 empty slot, which is 6 BT slots altogether, the length of the MIDI slot.

The above calculated value is 2 bytes less than the effective value, because most MIDI messages consist of 2 or 3 bytes, and it can happen that a 3-byte message follows after the 11th byte. The effective maximal length of a BT packet is such:

$$B_{TO\_TRANSMIT} = B_{1\_OUT} + 2 = 14 \text{ bytes.}$$

To be able to keep the timing constant both at high and low loads, let the length of the transmitted packets be constant, regardless of the number of useful bytes in them. The trasmission needs 2 more administrative bytes (a header and a footer). The one-byte header has to contain the number of the Outs, so that the merger can use this to identify the stream in which it has to insert the incoming bytes, while the one-byte footer indicates the end of the packet, which is implemented by using a byte not defined in the MIDI standard. So

$$B_{MERGER} = B_{TO\_TRANSMIT} + 2 = 16 \text{ bytes}$$

is the number of bytes that is to be transmitted during a MIDI slot. Since the DM1 packet can contain exactly 18 bytes of useful data and lasts for 1 BT slot, it is ideal to transmit this amount of information with this type of packet.

It is pretty plausible to use a Bluetooth module with an UART-type *Host Controller Interface* (HCI) because of its flexibility and simplicity, so from now on we show the timing values calculated specifically for UART based systems.

To be fair with the calculations by systems built with UART HCI BT modules it must be considered that the packet formatting needs 5 more bytes (1 byte ACL identifier, 2 bytes of connection handle, the ID of the master-slave physical attachment on-the-air, 2 bytes of flags and a packet length information, which is calculated without the 5 header bytes). These bytes will not be sent on the air, so we do not have to change the packet type. So 21 bytes have to be transmitted to the BT module, and the other module also will transmit this amount of bytes to its host.

With a 1 382 400 bits/s UART this process lasts for 152 µs (1 start bit + 8 data bits + 1 stop bit = 10 bits. T_{UART} = 10*21 / 1382400 = 152 µs), this is 24,3% of the length of the BT slot.

This calculation gives 506 µs for 2 Out-s with DM3 packets, which is still less than the length of a BT slot - which means that the timing will remain accurate.

In the case of 3 Out-s the situation is as follows: $B_{3\_OUT} = 20*S_{MIDI}T_{BT}$ = 40 bytes, because the broadcast packet must last 5 BT slots, as the length of the packet is 132 bytes which already needs the DM5 packet. The transmission lasts for 354 µs for the slaves and 998 µs for the master.

The length of the broadcast is 204 bytes in the case of having 4 Out-s in the system. The transmission times are 405 µs and 1519 µs, respectively.

After all there are no obstacles to use 5 Out-s in a piconet, but the broadcast packet cannot be realized using M packets here because of the 295 bytes to be transmitted. The transmission times are 463 µs and 2177 µs for the slaves and for the master. The latter is 3.48 in terms of BT slots, so the timing is still not vulnerable.

More than 5 Out-s are not applicable in the same piconet because of the bandwidth limits of the current Bluetooth technology.

Unfortunately in the respect of data protection and security the solutions above are pretty bad – nevertheless their delay is the best without doubt – because they transmit everything only once to the recipient. The

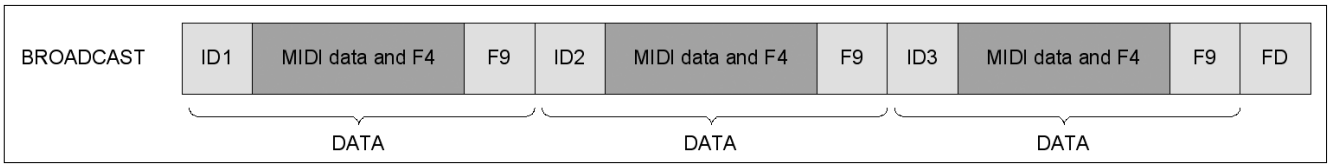


Figure 3. The structure of the broadcast packet

2/3 FEC coding improves the noise margin a bit, but the connection quality is still the function of the spatial placement and distance of the units. The most trivial method to minimize the packet loss probability is to use multiple transmissions.

Using 1 Out we can retransmit the data even 3 times; in this case the delay time is 16 ms, but it is still only 21 ms with 4 times of retransmission.

With 2 Out-s the maximum number of retransmissions is 2 (including the polling, the responses and the broadcasts); the delay time is then 18.75 ms.

Having 3 Out-s enables 2 transmissions for the polling, but does not allow us to retransmit the broadcast messages any more if we want to keep the timing. However, the broadcast can be repeated one more time if we use DH5 packets. With more than 3 Out-s none of the retransmission techniques can be applied with the data speed of the current Bluetooth standard.

The above are summarized in *Table 2*. We remark that using 1 Out port the throughput limit of Bluetooth enables 14 times of retransmissions without affecting the time stability of the MIDI slot.

The structure of the MIDI packets

Assuming maximum 3 Out-s *Figure 3.* shows the structure of the broadcast packet.

The DATA markings stand for the MIDI data of the each individual slave units. The termination mark of the data and the broadcast packet is the 0xF9 and 0xFD byte respectively, which are not defined in the MIDI standard. If the MIDI devices utilize the not defined 0xF4, 0xF5, 0xF9 or 0xFD MIDI bytes which are used for packet formatting, the packet headers, the timing and the structure of the protocol has to be modified to be able to transmit these bytes, too.

The maximum length of the broadcast packet is 238 bytes (DH5 packet, 3 Out-s, double transmission). The contents of the poll message are indifferent.

Reducing the overall latency by creating a scatternet

Using more masters simultaneously the latency can be decreased by distributing the Out units among the different masters. Unfortunately it is not easy to avoid masters transfer on the same frequency, although BT

Table 2. The latency of the data transfer – we suggest using the properties of the highlighted field

	No. of MIDI OUT-s	Response packet type (Slave)	Broadcast packet type (Master)	Latency (slots)	Transmitted slots (in order)	**Latency**	Physically transmitted MIDI bytes	Logically transmitted MIDI bytes
single transmit	1	DM1	DM1	6	1 poll, 1 response, 2 empty, 1 broadcast, 1 empty	**3.75 ms**	12	14
	2	DM3	DM3	14	2x1 poll, 2x3 response, 2 empty, 3 broadcast, 1 empty	**8.75 ms**	28	30
	3	DM3	DM5	20	3x1 poll, 3x3 response, 2 empty, 5 broadcast, 1 empty	**12.50 ms**	40	42
	4	DM3	DM5	24	4x1 poll, 4x3 response, 2 empty, 5 broadcast, 1 empty	**15.00 ms**	47	49
	5	DM3	DH5	28	5x1 poll, 5x3 response, 2 empty, 5 broadcast, 1 empty	**17.50 ms**	55	57
double transmit	1	DM3	DM3	18	2x1 poll, 2x3 response, 2 empty, 2x3 broadcast, 2x1 empty	**11.25 ms**	36	38
	2	DM3	DM5	30	2x2x1 poll, 2x2x3 response, 2 empty, 2x5 broadcast, 2x1 empty	**18.75 ms**	59	61
	3	DM3	DH5	38	2x3x1 poll, 2x3x3 response, 2 empty, 2x5 broadcast, 2x1 empty	**23.75 ms**	75	77
triple transmit	1	DM3	DM3	26	3x1 poll, 3x3 response, 2 empty, 3x3 broadcast, 3x1 empty	**16.25 ms**	51	53
	2	DM3	DH5	44	3x2x1 poll, 3x2x3 response, 2 empty, 3x5 broadcast, 3x1 empty	**27.50 ms**	86	88
4-times transmit	1	DM3	DM3	34	4x1 poll, 4x3 response, 2 empty, 4x3 broadcast, 4x1 empty	**21.25 ms**	67	69
5-times transmit	1	DM3	DM3	42	5x1 poll, 5x3 response, 2 empty, 5x3 broadcast, 5x1 empty	**26.25 ms**	83	85

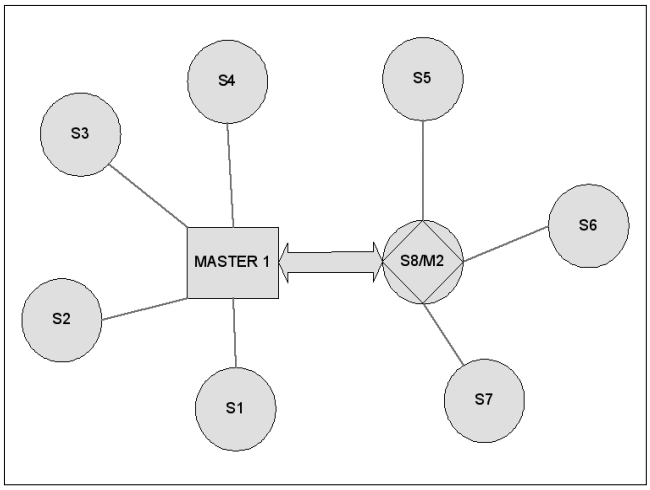


Figure 4. Scatternet with multifunctional device

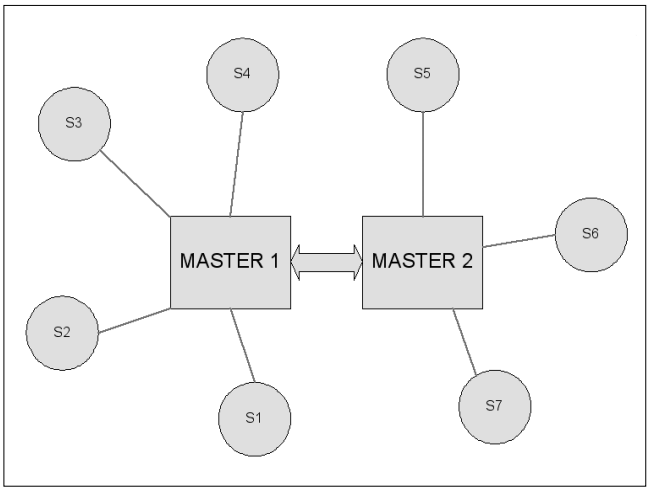


Figure 5. Scatternet by high-speed link

1.2 implements a method to avoid this, so there is even more need for multiple transmissions, which increases the latency at the same time. Creating arbitrary connection topologies is then realized by connecting the piconets, which is known as scatternet. It can be formed in more ways from piconets: a slave unit may operate as a master in the other piconet, or individual masters might be wired via a high-speed link.

Considering the MIDI implementation the first method is nowise adequate, because the multifunctional S8/M2 device (see *Figure 4*.) can only serve one of its functions at a time – synchronism cannot be achieved. However this raises another problem: what happens if the slave units of Out functionality are not balanced equally in the several piconets. As we could see in the discussion of the protocol timing, the latency increases and the parameters of the protocol implementation to be applied (e.g. number of retransmissions, type of BT packets) vary with the increase of the amount of the Out-s.

We find most expedient building a system where the end user does not have to reconfigure the whole system manually when putting a new master in operation and also does not have to set up the new connection topology in an uncomfortable and lengthy manner. When distributing the Out-s we have to strain after that each master gets the least Out-s possible and that the In-s receiving data from the same Out-s get into the same piconet, so that the least data have to be transmitted outside the piconet.

3. Conclusion

This paper made a proposal for a Bluetooth protocol conception and investigated its feasibility for implementing wireless MIDI connections. In spite that the implementations using Bluetooth did not succeed so far, the system described here fully exploits the more increasing throughput of Bluetooth. It allows acceptably safe data transfer while maintaining constant latency.

A system based on the contents of this paper can be easily extended by new units so that the overall latency can be decreased while the number of client units can be increased.

The most important conclusions for the feasibility of the system are:

(1) retransmission – at least for 2 times is needed to achieve a safer connection,
(2) a piconet may contain up to 3 MIDI Out-s, where only two Out-s are suggested to transmit data to the same piconet,
(3) there is no need for use of broadcast messages if there are no Out-s connected to a piconet, the DM3 packet is ideal for 1 or 2 Out-s, delay times of them are 16.25 ms and 18.75 ms, respectively, with a different number of retransmissions for each; while in the case of 3 Out-s, the DH5 packet should be used which results in 23.75 ms latency, and
(4) 3 masters at most can be tied together in the same area while maintaining a sufficient data reception probability if using Bluetooth 1.1. This does not apply for the 1.2 standard, which can implement piconets with arbitrary distribution of the 79 frequency channels.

References

[1] MMA MIDI Specifications, 1983-2003.
[2] Specification of the Bluetooth System, v1.1
[3] Specification of the Bluetooth System, v1.2
[4] J. Keniston, S. Sturdivant (2003):
Wireless MIDI Network Implemented Via Bluetooth
[5] R. Mettala: Bluetooth Protocol Architecture, v1.0
(Bluetooth White Paper Document # 1.C.120/1.0)

Introduction to Aspect-Oriented Programming

LÁSZLÓ LENGYEL, TIHAMÉR LEVENDOVSZKY

{lengyel, tihamer}@aut.bme.hu

Reviewed

Key words: aspect-oriented programming (AOP), crosscutting concerns

Aspect-oriented programming is a fortunate extension to the wide-spread object-oriented paradigm. In this paper we present the most important concepts of AOP based on the most widely used AspectJ approach. The problem of crosscutting concerns is introduced, and the facilities provided by AOP are enumerated as possible solutions. The most popular implementations (HyperJ, Composition Filters) are also mentioned briefly.

1. Introduction

Nowadays, the object-oriented programming (OOP) is the dominant paradigm of software engineering. The solutions provided by OOP can be applied to facilitate creating well-structured program as well as the code reuse. That is the reason for its wide adoption and its relative dominance. The concept behind the OO approach is that the program under development consists of autonomous entities, so-called objects, whose functionality is realized by the communications of these objects. This method, according to the experience, supplies a well-structured solution even for complex systems [1,2].

If a complex problem is decomposed into objects, the creation of the autonomous entities is focused along with the encapsulation of the data and the related operations. In this way, however, we have to ignore more important logical aspects of structuring and grouping, such as persistence or debug, which characteristically scattered across the code. This makes the software difficult to comprehend and maintain. These tangled but logically connected code parts that are scattered across different module are called *crosscutting concerns*.

An example for crosscutting concerns can be tracing the execution of a program. Distributed applications frequently write a log file, which helps debugging in case of an application error with collecting all the function calls and exceptions.

In order to write a log file, each class must contain program lines implementing the log functionality, usually scattered, whereas the code parts that perform logging are closely connected: they realize the same function.

As another possible example [3] the UML class diagram of a simple figure editor is illustrated in *Figure 1*. The *FigureElement* has two concrete descendants: the *Point* and the *Line*.

The decomposition into classes seems promising: both classes have a well-defined interface, and the data is encapsulated with the operations performed on them. However, the screen manager must be notified about the movement of each element. That demands that each function performing movements should notify the screen manager. The rectangle *DisplayUpdating* frames the functions that should implement this feature. Similarly, the rectangles *Point* and *Line* frame the functions implementing concerns related to them. It is worth noting that the *DisplayUpdate* square fits into none of the other rectangles in the figure, but it cuts across them.

Extending the OO facilities, Aspect-Oriented Programming (AOP) [4] offers a solution to the problem of crosscutting concerns. AOP divides the program code on the basis of the concerns that they contribute to the operation of the program. Approaching the problem in an AOP way, we can group the concerns into *aspects* implemented separately and independently of each other, and then an *aspect weaver* application joins these separate parts. Weaving is dependent on the particular AOP implementation it can happen either dynamically at run-time or statically at compiling time or after that.

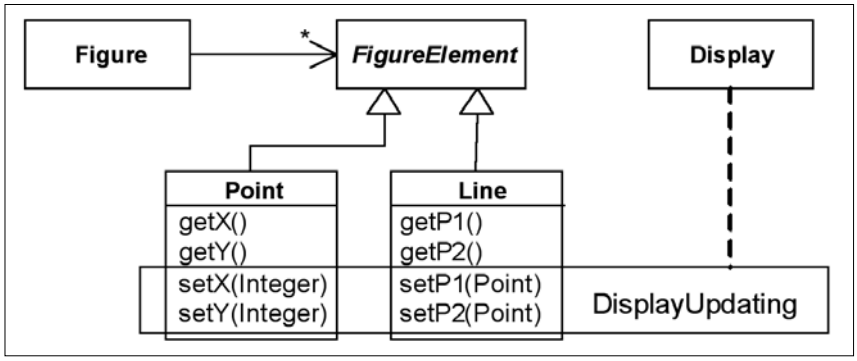


Figure 1. Crosscutting Concerns

If OOP is applied, the implementation of the crosscutting concerns are scattered across the system. However, if AOP mechanisms are used, the concern surrounded by the rectangle *DisplayUpdate* can be implemented using only one aspect. Besides these AOP facilitates thinking in aspects also on the design level along with realizing modularity.

The key point in modularization is that the program parts constituting one unit can be supplied in one physical unit as well. It is a general principle that the cohesion inside a module should be strong, and the modules should be loosely coupled. Using abstraction we can highlight the common traits of different elements. While the abstraction is rather vertical, separating the crosscutting concerns we can achieve structuring our system horizontally.

A programming paradigm or technology is mainly determined by the type of abstraction it uses. The frequent, repetitive code snippets or patterns mean the lack of abstraction facilities. The redundancy is unwanted, because a small change in the design may result that several, not connected module needs to be changed.

2. Crosscutting Concerns

Separating concerns means the ability that we need to identify and highlight those parts of the software which realize a concrete intention or goal. Separation of concerns primarily aims at decomposition of the software into parts that can be treated more easily and comprehensible. A natural question is how to accomplish this decomposition. What are the functions that should belong to a class or an aspect?

It is important to notice that the crosscutting is connected with a specific decomposition, since the crosscutting concerns cannot be separated completely. The basic design rule is to consider the fundamental concerns as a primary abstraction, and to implement them in classes and their extensions with aspects is done afterwards if needed.

Regarding the figure editor example, there are two important design concerns: representing the graphical elements and tracing the movement of them. The classes depicted in *Figure 1* represent the first concern. Each graphical class encompasses its inner data structure, which can be extended with aggregation and inheritance. The second concern, tracing the movement of the elements, should be implemented as separate classes, but the firs concern prevents this, because the functions realizing the movements are part of the graphical classes because of the encapsulation. The system could have been designed around the concern tracing the movements, but in that case the graphical functionality would have crosscut the tracing classes. Which solution is better?

The problem can be solved with the help of the dominant decomposition. The software – similarly to books – is written like text, as the book consists of chapters and paragraphs, the software has modules e.g. classes. The modules constituting the dominant decomposition contain uniform concerns and most of the time *they can be executed separately*. A dominant module cannot contain concern crosscutting several other modules. These are going to be crosscutting concerns.

Typical crosscutting concerns are synchronization, monitoring, buffering, transaction handling or context-sensitive error handling. Crosscutting concerns can be very high-level, e.g. security or the aspects of quality of service (QoS).

3. Aspect-Oriented Programming

The AOP paradigm was created in the mid-90s, and it has become an important area of research related to programming languages, so it is expected to gain more popularity.

It is apt to say that every programming language has a feature since Fortran, that facilitate to separate the concerns with subroutines. Subroutines are still useful constructs, and OOP cannot exist without block structures or structured programming. Similarly, AOP does not replace the technologies in use.

It happens quite often that the concerns cannot be realized by simple procedure calls, because a concern gets mixed with other structural elements, and becomes a fuzzy mess. The other disadvantage of subroutines that the programmers working on the caller component must be aware of the concerns, they must know how to include or use them. Besides the subroutines, AOP offers a call mechanism, where the developers of the caller components do not need to know about the extending concerns, namely, about calling the subroutine.

The two most important principle of AOP are the separation of crosscutting concerns and the modularity [5]. AOP has recognized that the boundary of modular units is rarely the same as the boundary of concerns. The modularization can be solved with certain concerns, but the code implementing the crosscutting concerns are scattered across the program, crosscutting the modular units. The main goal of AOP is to make it possible that the crosscutting concerns can be organized into autonomous modular units, thus it decrease the complexity of the software product (code and/or design), and the reuse, readability and maintenance of the program becomes simpler.

An aspect is a modular unit of implementation; it encapsulates a behavior, which affects several classes. Using AOP we implement the application in an arbitrary OO language (e.g. C++, Java or C#), then we deal with the crosscutting concerns, which means including the aspects. Applying the aspect weaver, finally, the executable application is built via combining the code and the aspects. As a result, the aspect becomes the part

of the implementation of several functions, modules, or objects, thus increasing the reuse as well as the maintenance facilities.

The weaving process is depicted in *Figure 2.* (on the next page). It is worth noting that the original code does not need not to know anything about the extending aspect, if one translates it without weaving, the result is the original application. If the aspects are included, the result is the application extended with the functionality of the aspects. It means that the original code need not be changed; the same program is used in both cases.

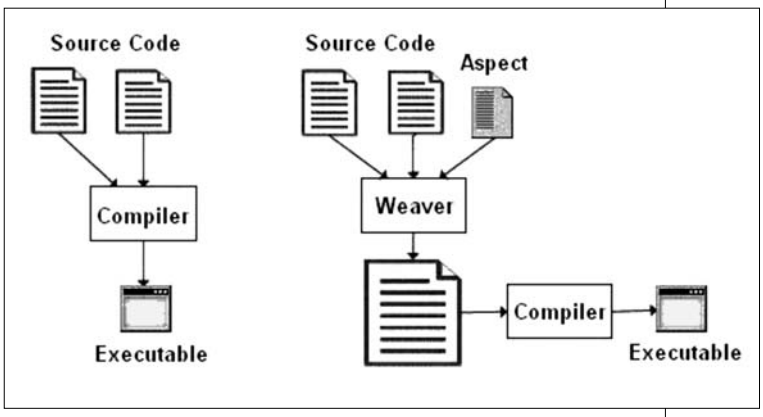


Figure 2. Aspect Weaver

AOP enhances, but does not replace OOP. It offers a different type of decomposition, in addition to classes, it introduces new element of modularization to realize each aspect separately from the classes in a different place. AOP is built on OOP, the objects and functions are not considered to be obsolete, aspects are meant to be used together with them.

The enhancement provided by AOP is that the entry point is declared by the function instead of the caller, which is not aware of calling the function. If the entry point of a function is given in the called part, it is woven into the code (obviously, it compiles to function call on the programming language level, but it is handled by the weaver automatically). Since the called code can be executed independently from its extension, as it has already been mentioned, this is a useful feature.

E.g. for memory paging, operating systems use a dirty bit to register the changes on the memory pages. If there is a change, the modified data must be saved to the storage. Using AOP techniques, handling the dirty bit can be separated from paging: the system can be executed with or without dirty bit handling, and the functions related to dirty bit handling are physically in the same place, instead of being scattered across the code according to the entry points.

Now a question arises, namely, how to provide where we want a piece of code to be called such that it does not appear in the caller part at all. The solution is the *join point*. The join points mean those well-defined places in the program, where the aspect interacts with the other parts of the system.

The join points mean the places of the program text or an execution point, so they can be divided into static or dynamic join points, respectively. Static join points are the first statement of a public function in case of the logging example, which results in a log file where the function call stack can be traced. Dynamic join points are connected with the events of the program execution like a method call (both inside of outside of the function), attribute query, throwing an exception, initializing a class or an object.

Aspect reuse is a fundamental result of AOP. The simple, small aspects facilitate the reuse of individual pieces of code more. Learning from the experiences and collecting aspects we can create aspect libraries. An apt question is how to deal with the large number of aspects and what notation should be used for them. This and similar questions are expected to be researched in the next few years.

A really important but open issue is the semantical correctness of the aspects. In case of component-based systems there has always been a question how to ensure the correct operation of the components. The AO approach offers far richer mechanisms than those provided by interfaces or message-based connections. Each aspect must thoroughly be examined from the point of specification and component test. If we use an aspect, there is no guarantee that we have the correct operation in every place where we reuse it afterwards. A way must be found to describe the operation of the aspects under specific circumstances.

Having presented the AOP, we briefly introduce the three most popular AOP implementations.

AspectJ

The environment AspectJ (www.aspectj.org) is a natural extension to Java: every Java program is an AspectJ program as well. The AspectJ introduces the following new programming language constructs for the AOP definitions [6,7]:

- *aspect:* A new programming unit, which encapsulates the crosscutting concerns. An aspect can contain the definitions of pointcuts, advices and introductions. Similarly to classes, they can have methods and data members.
- *pointcut:* It defines a set of dynamic join points with the help of a logical expression. These are called pointcut descriptors. Pointcuts can be parameterized, the objects of the pointcut environment can be passed to the advice in the parameters.
- *advice:* It is a programming unit similar to methods. Advices are always associated with a pointcut. Their body contains the behavior that should be executed in the join point described by the pointcut.
- *introduction:* Introductions help to define new data members and methods in classes. Here the join point is the class.

In order to decide the runtime order in which the elements of the aspects associated with the same join point are executed, precedence relations have been established within the aspects [8].

Classes and aspects are not on the same level in AspectJ [9,10].

Whereas classes can be regarded as autonomous entities, aspects can be interpreted along with the classes whose crosscutting cede they contain. The aspects can be considered that they contain the modifications of the original program code. Therefore aspects cannot be compiled alone, their reuse is possible on the source level only. The current AspectJ implementation works at compile-time only and the base program code is also necessary [11].

Hyperspaces – HyperJ

The hyperspace approach (www.research.ibm.com/hyperspace/index.htm) is based on the multidimensional separation of concerns. According to this principle there are several concerns of different types in the software, while the currently popular languages and methods facilitate the decomposition driven by only one concern. This phenomenon is called the tyranny of the dominant decomposition. The basis of the dominant decomposition is classes (OO languages), functions (functional languages) and rules (rule-based programming languages).

Hyperspaces facilitate identifying explicitly an arbitrary dimension of the concerns in the arbitrary phase of the development process. The hyperspace model uses the following definitions:

- *hyperspace:* It is for identifying the concerns. Hyperspace means the set of the software building blocks. E.g. classes in an OO environment, a method, or a data member. The hyperspace organizes these elements into a multidimensional matrix. In the imaginary coordinate system of the hyperspace the concerns are the dimensions, and the coordinates are the specific concerns within a dimension. The coordinates of the units in the hyperspace define concern that the given unit describes.
- *hyperslice:* It encapsulates the concerns. The units belonging to the same concerns are placed on the same hyperslice. Defining hyperslices we can encapsulate the units related to the concerns.
- *hypermodule:* It serves as a basis to integrate the concerns. A hypermodule encompasses a set of the hyperspaces to be integrated as well as the integrating relations describing the way of the integration and the relations between the hyperslices.

The tool HyperJ is a Java implementation for multidimensional separation of concerns [12]. The HyperJ performs the integration on the compiled hyperslice packages, not on the source code, thus one can remodularize the already existing applications for reuse. The join points are static, their definitions are contained by the specification of the hypermodule.

Composition Filters

Since in the OO languages the behavior is determined by the messages passed between the objects, a large scale of the behavioral modifications can be achieved by manipulating the incoming and outgoing messages (typically the function calls) of the objects.

In the model of composition filters (http://trese.cs.utwente.nl/composition_filters/) the manipulation and analysis of the messages are performed by filters. The model expresses the crosscutting concerns as the modular and orthogonal extensions of the objects. The modularity is ensured by the well-defined interfaces of the filters, and they inherently independent from the implementation of the objects [13,14]. The filters are orthogonal to each other, because in their specification there is no reference to other filters.

4. AOP and crosscutting constraints

Aspect-Oriented Software Development (AOSD) [4] is a new technology that has introduced the *separation of concerns* (SoC) in software development. The methods of AOSD facilitate the modularization of crosscutting concerns within a system. Aspects may appear in any stage of the software development lifecycle (e.g. requirements, specification, design, implementation etc.). Crosscutting concerns can range from high-level notions of security to low-level notions like caching and from functional requirements such as business rules to non-functional requirements like transactions. AOSD has started at the programming level of the software development life-cycle and the last decade several aspect-oriented programming languages have been introduced.

Aspect-oriented programming eliminates the crosscutting concerns in the programming language level, but the aspect-oriented techniques must be applicable on a higher abstraction level as well to solve this issue. In [15] an aspect oriented approach is introduced for software model containing constraints where the dominant decomposition is based upon the functional hierarchy of a physical system.

The modularization of crosscutting concerns is also useful in model transformation. Model transformation means converting an input model available at the beginning of the transformation process to an output model or to source code. Models can be considered special graphs; simply contain nodes and edges between them. This mathematical background makes possible to treat models as labeled graphs and to apply graph transformation algorithms to models using graph rewriting. Therefore a widely used approach to model transformation applies graph rewriting [16] as the underlying transformation technique, which is a powerful tool with a strong mathematical background.

The atoms of graph transformation are rewriting rules, each rewriting rule consists of a left hand side

graph (LHS) and right hand side graph (RHS). Applying a graph rewriting rule means finding an isomorphic occurrence (match) of the LHS in the graph the rule being applied to (host graph), and replacing this subgraph with RHS. Replacing means removing elements that are in the LHS but not in the RHS, and gluing elements that are in the RHS but not in the LHS.

In general, graph rewriting rules parse graphs only by topological concerns, but they are not sophisticated enough to match a graph with a node which has a special property or there is a unique relation between the properties of the parsed nodes.

In case of diagrammatic languages, such as the *Unified Modeling Language* (UML), the exclusive topological parsing is found to be not enough. To define the transformation steps in a more refined way – beyond the topology of the graphs – additional constraints must be specified which ensures the correctness of the attributes among others.

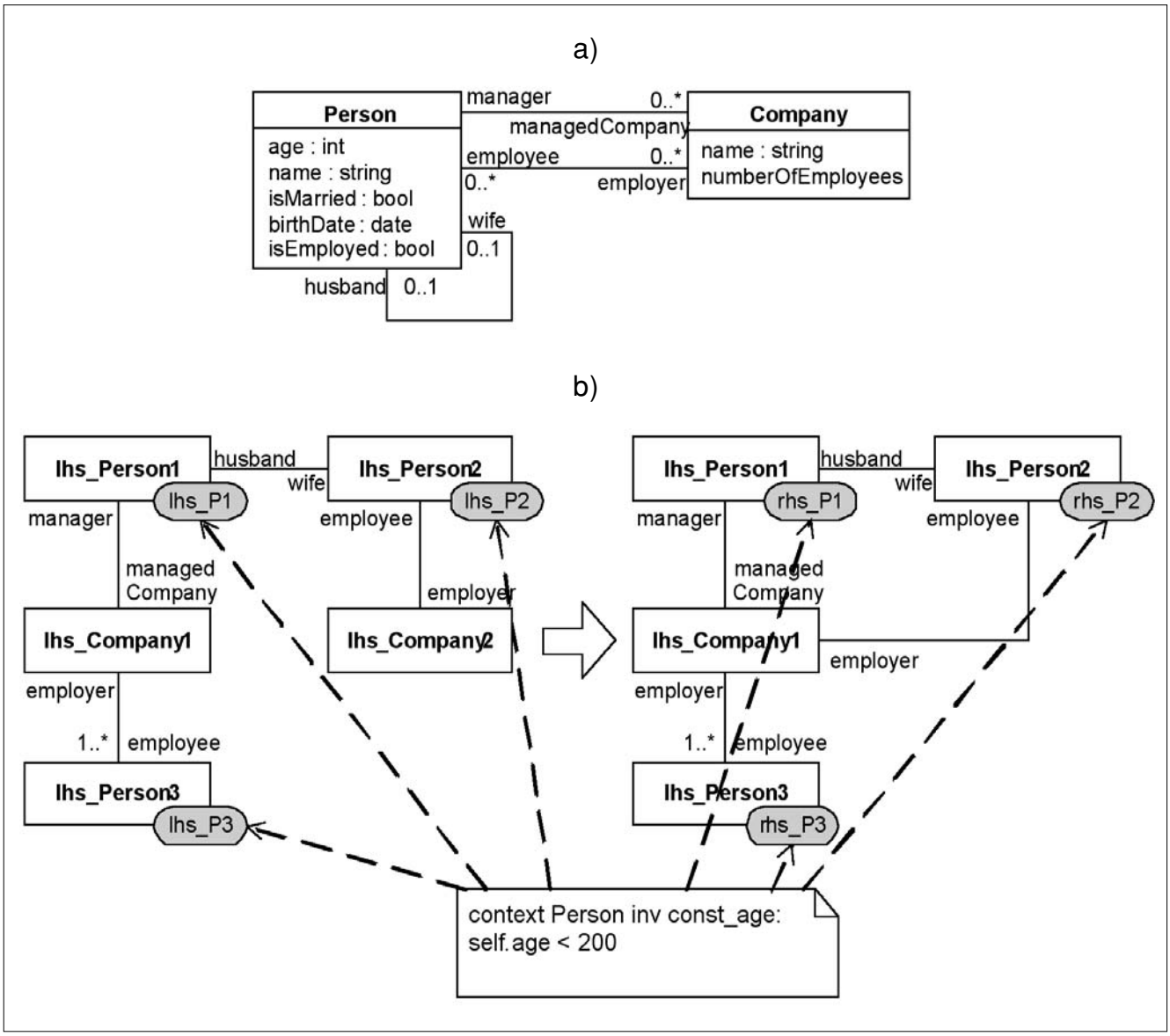
Dealing with constraints provides a solution for the unsolved issues, because topological and attribute transformation methods cannot perform and express the problems, which can be addressed by constraint validation.

The use of the constraints in graph transformation rules and in graph rewriting is found to be useful. Often, the same constraint is repetitiously applied in many different places in a transformation.

E.g. we have a transformation which modifies the properties of *Person* type objects and we would like the transformation to validate that the *age* of a *Person* is always under 200 *(Person.age < 200)*. It is certain that the transformation preserves this property if the constraint is defined for all rewriting rule element whose type is *Person (Figure 3/b)*.

It means that the constraint can appear several times, and therefore the constraint crosscuts the whole transformation, its modification and deletion is not con-

Figure 3. A sample metamodel and a transformation step with a crosscutting constraint
a) Metamodel b) Transformation step



sistent because such an operation has to be performed on all occurrence of the constraint. Besides this often it is difficult to reason about the effects of a complex constraint when it is spread out among the numerous nodes in rewriting rules.

It would be beneficial to describe a common constraint in a modular manner, and to designate the places where it is to be applied. We need a mechanism to separate this concern. Having separated the constraints from the pattern rule nodes, we need a weaver method which facilitates the propagation (linking) of constraints to transformation step elements.

It means that using separation and weaver method we can manage constraints using AO techniques: Constraints can be specified and stored independently of any graph rewriting rule or transformation step node and can be linked to the rewriting rule nodes by the weaver.

To summarize the main idea of the AO constraints, we can say that one can create the constraints and the rewriting rules separately, and with the help of a weaver constraints can be propagated optional time to the rewriting rule nodes contained by the transformation steps. Therefore constraints are similar to the aspects in AOP.

5. Conclusions and future work

AOP is a language-independent construct, a concept above the implementations. In fact, it can remedy the shortcomings of the programming languages (not only OO) with a simple and hierarchical decomposition.

The AOP concepts have been implemented in several programming languages: C, C++, Java, Perl, Python, Ruby, SmallTalk and C#. The research community targets Java the most, thus the most sophisticated tools and environments are available in this language.

In the field of software engineering the long-term issues of the software lifecycle play a more important role nowadays. These include the problems of simplifying the development, maintenance, being able to accommodate to changes or reuse. AOP helps to achieve these goals with ensuring a model more flexible and working on a higher abstraction level than the OO paradigm.

The number of publications is quite large. As a starting point we recommend the special issue of CACM devoted to the topic [17] and the web sites of each implementation.

References

[1] Czarnecki, K. and Eisenecker, U.W.: Generative Programming: Methods, Tools and Applications. Addison Wesley, Boston, 2000.

[2] Dijkstra, E.W.: A Discipline of Programming. Prentice-Hall, 1976.

[3] Tzilla Elrad, Mehmet Aksit, Gregor Kiczales, Karl Lieberherr, Harold Ossher: Discussing Aspects of AOP, CACM Vol. 44, Issue 10 (October 2001)

[4] Aspect-Oriented Software Development, http://www.aosd.net/

[5] Tzilla Elrad, Robert E. Filman and Ataf Bader: Aspect-oriented Programing, CACM Vol. 44, Issue 10 (October 2001)

[6] Gregor Kiczales, Erik Hilsdale, Jim Hungunin, Mik Kersten, Jeffrey Palm and William G. Griswold: An Overview of AspectJ. J. Lindskov Knudsen (Ed.): Proceedings of the 15th ECOOP, Budapest, 2001, pp.327–353.

[7] Gregor Kiczales, Erik Hilsdale, Jim Hugunin, Mik Kersten, Jeffrey Palm and William G. Griswold: Getting started with AspectJ, CACM Vol. 44, Issue 10 (October 2001)

[8] Kiczales, G., et al.: An overview of AspectJ. In Proceedings of the 15th European Conference on OOP (ECOOP). Springer, 2001.

[9] The AspectJ Programming Guide, www.aspectj.org

[10] Bill Griswold, Erik Hilsdale, Jim Hugunin, Wes Isberg, Gregor Kiczales, Mik Kersten: Aspect-Oriented Programming with AspectJ, http://www.aspectj.org

[11] The AspectJ Primer, www.aspectj.org/doc/primer.

[12] Peri Tarr, Harold Ossher: Hyper/J User and Installation Manual, http://www.research.ibm.com/hyperspace http://www.math.klte.hu/~espakm/GOF/hires/Pictures/mvc.gif#

[13] Mehmet Aksit, Bedir Tekinerdogan: Solving the modeling problems of object-oriented languages by composing multiple aspects using composition filters, 1998.

[14] Mehmet Aksit, Lodewijk Bergmans: Software evolution problems using composition filters. ECOOP 2001, Budapest.

[15] Jeff Gray, Ted Bapty, Sandeep Neema: Aspectifying Constraints in Model-Integrated Computing, OOPSLA Workshop on Advanced Separation of Concerns in Object-Oriented Systems, Minneapolis, MN, October 2000

[16] Rozenberg (ed.), Handbook on Graph Grammars and Computing by Graph Transformation: Foundations, Vol. 1., World Scientific, Singapore, 1997.

[17] Communications of the ACM, Vol. 44, Issue 10 (October 2001)

Novel techniques for assessing resource requirements in packet-based networks

MÁTYÁS MARTINECZ, JÓZSEF BÍRÓ, ZALÁN HESZBERGER

martinecz@tmit.bme.hu

The authors are supported by Inter-University Centre of Telecommunications and Informatics (www.etik.hu)

Reviewed

Key words: equivalent capacity, QoS, call admission control

The lack of quality of service (QoS) guarantees is the classic problem of packet switching networks. Despite the access technologies (e.g. DSL) providing sufficient transmission speed are already available, without such QoS guarantees the rapid spread of novel, value-added services can not be imagined. In this article a novel technique capable to approximate the minimum bandwidth that should be provided for an aggregated network traffic flow in order to maintain a predefined QoS level is introduced. This new method can form the basis of load control (e.g. call admission control) algorithms to be applied in future packet-based networks.

1. Introduction

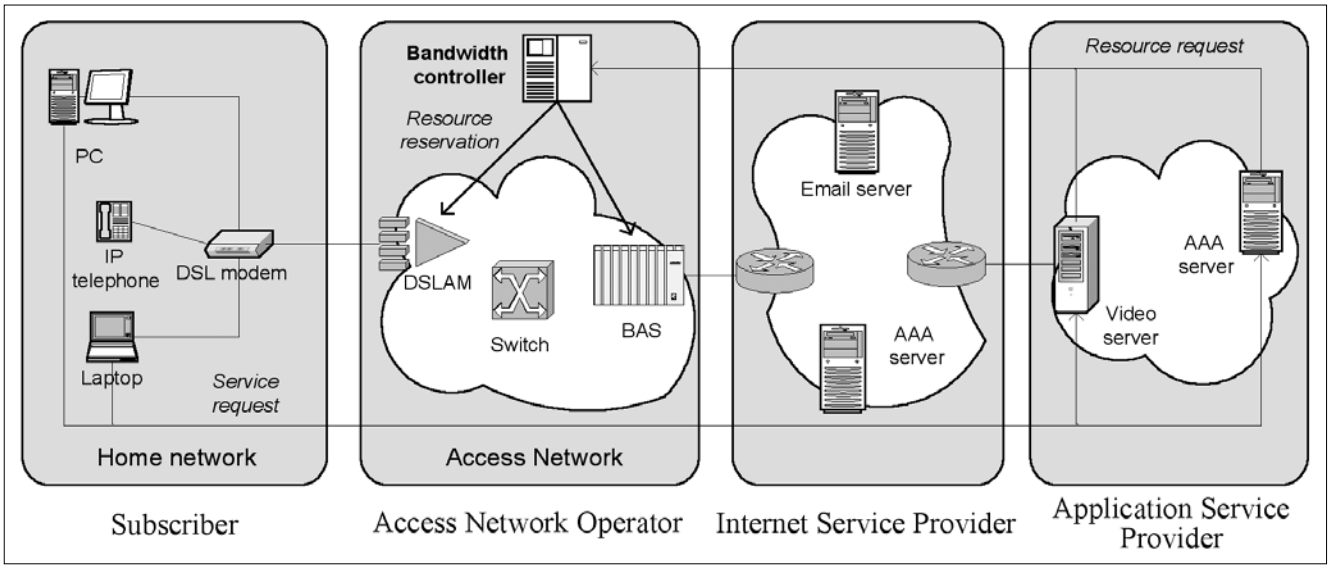
The number of DSL subscribers increases rapidly these days. This fact can be explained by the reasonable price and the relatively high reachable data rate this type of access technology offers. The reason of low price is that for the DSL access technology the already-in-use symmetric copper wires can be used by exploiting their higher (>144 kHz) frequency domain. As these copper wires can already be found at telephone users, in many cases the installation of DSLs may be the cheapest and best choice.

Amongst the services provisioned through DSLs the Internet access is the most popular [1]. The digital subscriber lines provide sufficient data rate for the majority of currently available services offered via the Internet. The previously available slow-speed access technologies severely hindered the development and spread of web applications. With the show-up of DSLs however the evolution of modern, broadband network services gained a new momentum. The spread of these premium applications (e.g. VoIP, VoD) is also encouraged by access network operators as they may attract new subscribers into their domain.

One of the gravest problems of TCP/IP based packet switching networks is the lack of transmission quality guarantees. Without these guarantees however the introduction and spread of value-added services is unimaginable. QoS guarantees and network management algorithms are primarily needed where resources may be scarce: in the local loop and the access network.

The quality of data transmission depends on the actual load of the network, which may be characterized by the saturation probabilities or packet loss ratios measured over the links of the network domain. The former metric (in case there are no buffers attached to links) is the probability of the instantaneous data rate

Figure 1. QoS guaranteed service provisioning with a bandwidth controller located in the access network



of the aggregated traffic flowing through a link exceeds its capacity. Unfortunately the link saturation probability does not tell anything about the amount of lost information, so it is more useful to prescribe the desired packet loss ratio, which is the ratio of lost and sent packets.

In this article a novel technique capable to approximate the expected load status of link while requiring only few a priori parameters is introduced. This approximation can be used by a bandwidth controller to supervise a network domain and making efficient and reliable admission control decisions.

The formulae to be presented approximate the required bandwidth need of a certain aggregated traffic flow in contrast with those that determine the expected level of QoS for a given link capacity. The advantage of our method is that it is enough to periodically refresh the actual amount of required bandwidth by background computations, while in the other case the expected QoS level has to be checked each time a new service request arises, which of course considerably slows down making admission decisions.

The rest of this article is organized as follows. In the next part the applied mathematical model is explained briefly. In the third section techniques capable to approximate the moment generating function of the aggregated traffic's rate distribution are presented. In the fourth part methods to convert QoS level approximations into bandwidth requirement values are introduced. In the fifth section the efficiency of previously and newly developed methods are compared through numerical examples. Our concluding remarks can be found in the last section.

2. QoS metrics and their approximation in packet-based networks

For the approach to the problem outlined in the introduction we used the popular BFFM (Bufferless Fluid Flow Multiplexing) framework. As in this model there are no buffers that may reduce the packet loss ratio it can be used to approximate important QoS metrics in a conservative manner.

Let us suppose that n fluid flows are aggregated on a link with capacity C. Let X_i be a random variable denoting the instantaneous data rate of the ith stationary flow. Let us suppose that for each source a p_i peak data rate can be determined, that is $0 \le X_i \le p_i$. Let X be a random variable denoting the instantaneous data rate of the aggregated traffic flow: $X = \sum_{i=1}^{n} X_i$.

Thus the link saturation probability can be defined as:

$$P_{sat} \stackrel{def}{=} P(X > C) \tag{1}$$

This probability means the fraction of time when the instantaneous data rate of the aggregated traffic exceeds the link capacity and so information loss occurs. This metric can be determined relatively easily, but may be useful only for the network operators, as the saturation probability does not give any reliable information regarding the amount of lost data. It is easy to imagine that beside the same saturation probability the number of lost packets may totally differ. Thus the level of users' satisfaction should be characterized with the packet loss ratio instead. It is by definition:

$$PLR \stackrel{def}{=} \frac{E\left[(X-C)^{+}\right]}{E[X]} \tag{2}$$

where $E[.]$ is the expected value operator, and $(X-C)^{+}=\max(X-C,0)$. So in other words the packet loss ratio can be computed as the expected value of the instantaneous data rates exceeding the link capacity (and so causing packet loss) divided by the mean instantaneous data rate of the aggregated flow.

The call admission decision is based on the relation of the expected and prescribed level of the QoS metric:

$$P(X > C) \le e^{-\gamma} \quad \text{vagy} \quad \frac{E\left[(X-C)^{+}\right]}{E[X]} \le e^{-\gamma} \tag{3}$$

Practically it is more tractable to compare the equivalent capacity of the aggregated traffic to the link capacity. The equivalent capacity is the minimum required bandwidth that the aggregated traffic needs for attaining the predefined QoS level. The definition of *equivalent capacity* can be written in the following forms in case the guaranteed QoS level is composed in terms of saturation probability or packet loss ratio:

$$C_{equ,sat} \stackrel{def}{=} \inf\left\{C: P_{sat} \le e^{-\gamma}\right\} \quad \text{or}$$

$$C_{equ,PLR} \stackrel{def}{=} \inf\left\{C: PLR \le e^{-\gamma}\right\} \tag{4}$$

For the approximation of the expected link saturation probability or packet loss ratio the Bahadur-Rao extension of the well-known Chernoff bound can be used.

The Bahadur-Rao approximation of the given QoS metric is more accurate than the Chernoff bound, however it is not necessarily conservative (i.e. it may underestimate the true value) [5,6]:

$$P(X > C) \approx \frac{1}{s^{*}\sqrt{2\pi\sigma^{2}(s^{*})}} \exp(\Lambda_X(s^{*}) - s^{*}C) \quad \text{or} \tag{6}$$

$$PLR \approx \frac{1}{M(s^{*})^{2}\sqrt{2\pi\sigma^{2}(s^{*})}} \exp(\Lambda_X(s^{*}) - s^{*}C) \tag{7}$$

where $\Lambda_X(s)$ is the logarithmic moment generating function *(LMGF)* of X,

$$M \stackrel{def}{=} E[X], \quad \sigma^{2}(s) = \frac{\partial^{2}}{\partial s^{2}}\Lambda_X(s), \quad s^{*} = \arg\inf_{s}\left\{\Lambda_X(s) - sC\right\}$$

Abbreviations

BFFM – Bufferless Fluid Flow Multiplexing
DSL – Digital Subscriber Line
LMGF – Logarithmic Moment Generating Function
PLR – Packet Loss Ratio
QoS – Quality of Sevice

It can be seen that for the presented approximations (6) and (7) the LMGF of the distribution function of X is needed. The LMGF can be computed if all the moments of X are known, which is usually not the case. To overcome this problem three methods for approximating the moment generating function of X are presented in the next section. These techniques are easy-to-implement as they require only three parameters: the number of flows, the peak data rates of flows and the mean data rate of the aggregated flow.

3. Parsimonious upper bounds of the moment generating function

The first approximation method with which an upper bound for the moment generating function of X can be determined is a corollary of the results published by Hoeffding in 1963 [2]. Let $X_i, i=1...n$ denote independent, bounded random variables, for which

$$X=\sum_{i=1}^{n} X_i,\quad M \stackrel{def}{=} E[X],\quad 0 \le X_i \le p_i.$$

Then for $s>0$

$$G_X(s) \le \exp(sM)\exp\left(\frac{s^2\sum_{i=1}^{n} p_i^2}{8}\right), \tag{8}$$

where $G_X(s)$ is the moment generating function of X.

Using Hoeffding's results Heszberger et al [3] formed the following conservative bound which can be applied to bound the moment generating function of the sum of bounded ($0 \le X_i \le p_i$) random variables ($X=\sum_{i=1}^{n} X_i$) :

$$G_X(s) \le \left(\frac{M+\sum_{k=1}^{n}\frac{p_k}{e^{sp_k}-1}}{n}\right)^n \prod_{i=1}^{n}\left(\frac{e^{sp_i}-1}{p_i}\right). \tag{9}$$

The already presented two bounds are based on the results of Hoeffding, however another approach may also be used for obtaining an upper bound for the moment generating function. For the construction of this third bound the concept of a certain type of stochastic ordering of random variables will be used. Let us assume that we have two random variables X and Y, whose distribution functions are denoted by F_X and F_Y, respectively. Then X is said to be smaller than Y with respect to increasing convex ordering [4], written as $X<_{icx}Y$, if the condition

$$\int_{-\infty}^{\infty}\phi(x)dF_X(x) \le \int_{-\infty}^{\infty}\phi(x)dF_Y(x)$$

holds for each increasing convex function $\phi(x)$, for which the integral exists.

From the definition it can be deduced that if $X<_{icx}Y$, then for $s>0$, $G_X(s) \le G_Y(s)$ holds. This can easily be verified by substituting $\phi(x)$ with e^{sx}.

Using the following lemma a new approximation for the upper bound of the moment generating function can be constructed [4]. Let $X_1^{onoff},...,X_n^{onoff}$ random variables denote n independent, heterogeneous (i.e. non-uniformly bounded) on-off sources whose peak data rates are $p_1,...,p_n$, and mean data rates are $m_1,...,m_n$, respectively. Let $Y_1^{onoff},...,Y_{n_Y}^{onoff}$ random variables denote n_Y independent homogeneous on-off sources, whose peak data rates are identically $p=max(p_i, i=1,...,n)$, and $n_Y=\text{int}\left\{\sum_{i=1}^{n} p_i/p\right\}$ (i.e. the upper integer value of the expression between the braces), their mean data rates are identically $m=\sum_{i=1}^{n} m_i/n_Y$. Then

$$X_{onoff} <_{icx} Y_{onoff},\qquad X_{onoff} \stackrel{def}{=} \sum_{i=1}^{n} X_i^{onoff},\qquad Y_{onoff} \stackrel{def}{=} \sum_{i=1}^{n_Y} Y_i^{onoff}.$$

Using this lemma and the consequence of the definition of increasing convex ordering the upper bound of the moment generating function of the X can be written as follows [8].

Let $X_i, i=1...n$ denote independent, bounded random variables,

$$X=\sum_{i=1}^{n} X_i,\ M \stackrel{def}{=} E[X],\quad 0 \le X_i \le p_i.$$

Then for $s>0$

$$G_X(s) \le \left(1-\frac{M\left(1+e^{sp}\right)}{n_Y p}\right)^{n_Y}. \tag{10}$$

From now on the logarithms of the moment generating function bounds (8), (9) and (10) (i.e. the corresponding LMGFs) will be denoted by $\tilde{\Lambda}_{X,hoe}(s)$, $\tilde{\Lambda}_{X,ih}(s)$ and $\tilde{\Lambda}_{X,so}(s)$ respectively.

4. Direct equivalent capacity estimation methods

By putting the previously introduced moment generating function bounds into formulae (6) and (7), an upper bound of the expected QoS level (saturation probability or PLR) can be obtained. This value then can be compared with the prescribed QoS level – as it was indicated in (3) – and the admission decision can be made according to the result of this comparison.

If we take another look on (6) and (7) we see that the original Bahadur-Rao formulae contain not only the LMGF, but also the second derivative of the LMGF.

Investigations show that as the exact moment generating function is not known (only an upper bound of it can be obtained), for its second derivative only a very imprecise approximation can be given, and this eventually makes the Bahadur-Rao formulae inapplicable. Thus it is desirable to eliminate the second derivative from the formulae somehow. It can be managed by using the results of Montgomery and de Veciana [7]:

$$P_{sat} \approx \exp\left(-I-\frac{1}{2}\log 4\pi I\right), \tag{11}$$

$$PLR \approx \exp\left(-I-\frac{1}{2}\log 4\pi I-\log s^* M\right), \text{ where} \tag{12}$$

$$I=-\inf_s\{\Lambda_X(s)-sC\},\quad s^*=\arg\inf_s\{\Lambda_X(s)-sC\}.$$

	n_1	m_1 [kbit/s]	p_1 [kbit/s]	n_2	m_2 [kbit/s]	p_2 [kbit/s]	P/M
M1	100	51	64	10	200	500	2,24
M2	100	51	64	1000	4,8	5,8	1,34

Table 1. Characteristics of the investigated traffic mixes

It was mentioned earlier, that it is more tractable to compute the equivalent capacity of an aggregated flow instead of the expected QoS level, because the equivalent capacity need to be refreshed only periodically while the expected value of the appropriate QoS metric should be recomputed each time a new service requests arrives. In case the equivalent capacity is tracked, a new flow can be admitted if its peak data rate plus the actual equivalent capacity of the aggregated flow is less than or equal to the link capacity.

If formula (11) or (12) is used in the appropriate part of formula (4), an indirect method for computing the equivalent capacity can be obtained. However, in this case a double optimization should be performed (with respect to parameters *s* and *C*), which considerable increases the computational complexity of this method.

To overcome this problem we have developed direct formulae which are capable to determine the equivalent capacity in one step. These can be written in the following forms (13 and 14):

$$\widetilde{C}^{B-R}_{equ,sat} \overset{def}{=} \inf_{s>0}\left\{\frac{\widetilde{\Lambda}_X(s)}{s}+\frac{\gamma}{s}-\frac{\gamma\log 4\pi\gamma}{s(1+2\gamma)}\right\}$$

$$\widetilde{C}^{B-R}_{equ,WLR} \overset{def}{=} \inf_{s>0}\left\{\frac{\widetilde{\Lambda}_X(s)+\gamma-1+\log M+\frac{2\gamma}{1+2\gamma}\log\frac{1+2\gamma}{4M\sqrt{\pi\gamma}^{\frac{3}{2}}}}{-\frac{1}{M}+s}\right\}$$

where $\tilde{\Lambda}_X(s)$ can be any appropriate approximation of $\Lambda_X(s)$ (e.g. the bounds presented in Section 3 are such). A more detailed discussion of these new results can be found in [8].

5. Numerical investigations

In this section the comparative analysis of the presented moment generating function bounds and equivalent capacity estimators will be carried out through numerical examples. For this investigation let us define a two-class, on-off traffic mix. The numbers of sources within the classes are represented by n_1 and n_2, respectively. The peak and mean data rate of the sources belonging to the same class are identical, these are denoted by m_i and p_i, $i \in \{1,2\}$. The important characteristics of the investigated traffic mixes are summarized in *Table 1*.

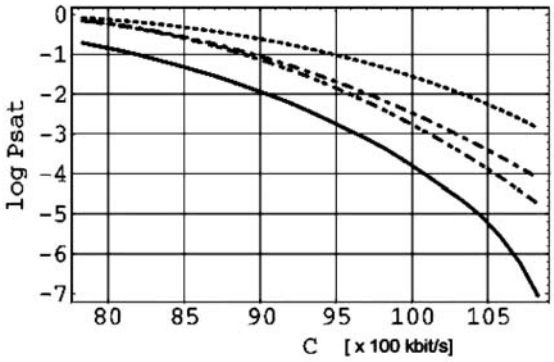


Figure 2.
Link saturation probability estimations, M1

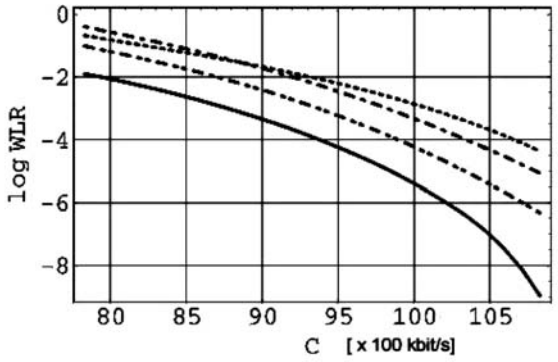


Figure 3.
Packet loss ratio estimations, M1

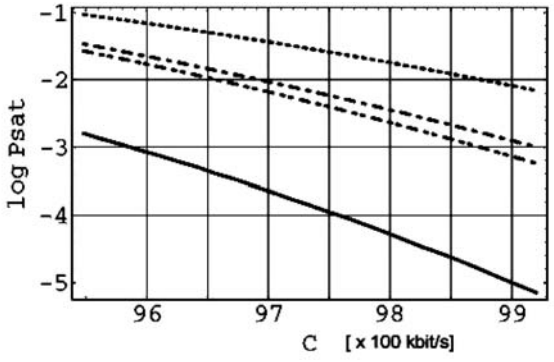


Figure 4.
Link saturation probability estimations, M2

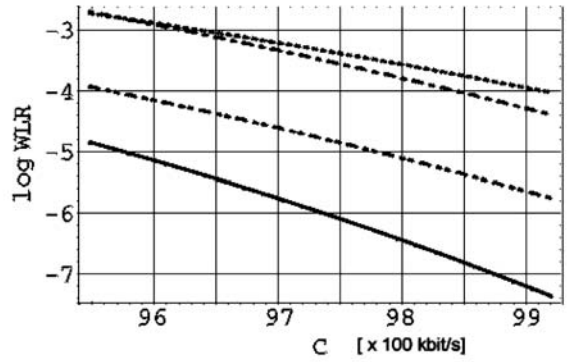


Figure 5.
Packet loss ratio estimations, M1

The first traffic mix (M1) can be considered as the aggregate of uncompressed voice and compressed video flows, while the second traffic mix (M2) resembles to the aggregation of compressed and uncompressed voice flows. The difference between the two mixes lies in the difference of the aggregate peak to mean ratio (shown in the last column of Table 1).

On *Figures 2-5* (on the previous page), the 10-based logarithms of the exact and approximated values of the link saturation probability or packet loss ratio are drawn as a function of the link capacity *C*. As the presented bounds give applicable results in the

$M < C < P$ ($P \stackrel{def}{=} \sum_{i=1}^{n} p_i$) interval,

only a part of the *(M,C)* interval is plotted. The exact values are drawn with continuous, while the bounds are drawn with dotted ($\tilde{\Lambda}_{X,hoe}(s)$), dash-dotted ($\tilde{\Lambda}_{X,ih}(s)$) and dash-dot-dotted ($\tilde{\Lambda}_{X,so}(s)$) lines.

On the figures it can be seen that in most cases the $\tilde{\Lambda}_{X,hoe}(s)$ bound is the less accurate, while the other two bounds' accuracy is acceptable. The vertical and horizontal distances between the curves usually increase with increasing γ (as the prescribed QoS level gets more stringent). The difference between the bounds of $\tilde{\Lambda}_{X,ih}(s)$ and $\tilde{\Lambda}_{X,so}(s)$ is sometimes negligible, however the computational complexity of those are fairly different: the stochastic ordering based bound can be obtained more easily. The application of the $\tilde{\Lambda}_{X,hoe}(s)$ bound can be recommended only if the computational complexity is the most important factor.

The performances of the equivalent capacity estimator formulae have also been compared. The numerical analysis was carried out the following way. First the exact values of the saturation probability and the packet loss ratio were determined for a given *C* value. Then from the $P_{sat}=e^{-\gamma}$ or $PLR=e^{-\gamma}$ formulae the corresponding γ values were determined. These γ values and the previously obtained $\tilde{\Lambda}_{X,hoe}(s)$, $\tilde{\Lambda}_{X,ih}(s)$ and $\tilde{\Lambda}_{X,so}(s)$ bounds were finally substituted into (13) or (14). The relation between the exact (i.e. in this case the link capacity *C*) and approximated value of the equivalent capacity has been then investigated.

On *Figures 6-7,* the $(\tilde{C}^{B\text{-}R}_{equ,sat}-C)/C$ relative error was drawn for M1 and M2 traffic mixes.

The equivalent capacity estimation for which $\tilde{\Lambda}_{X,hoe}(s)$ bound was used is plotted with continuous line, while the dotted and dash-dotted lines refer to the equivalent capacity approximations for which $\tilde{\Lambda}_{X,ih}(s)$ or $\tilde{\Lambda}_{X,so}(s)$ was used. It can be seen that the $\tilde{\Lambda}_{X,hoe}(s)$ based approximation severely underestimates, while for traffic mix M2 the $\tilde{\Lambda}_{X,so}(s)$ based approximation partly underestimates the exact equivalent capacity.

On *Figures 8-9,* the $(\tilde{C}^{B\text{-}R}_{equ,WLR}-C)/C$ relative error was drawn for the two traffic mixes. The equivalent capacity approximation for which $\tilde{\Lambda}_{X,hoe}(s)$ bound was used is

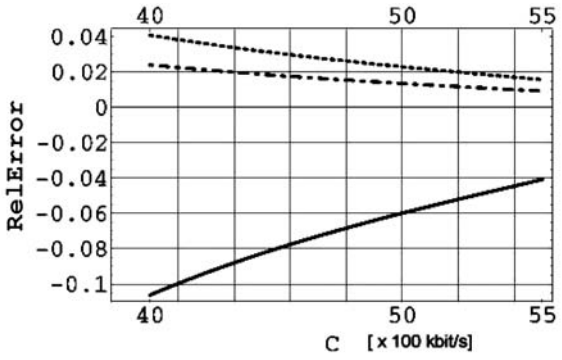


Figure 6.
The relative error of $\tilde{C}^{B\text{-}R}_{equ,sat}$, M1

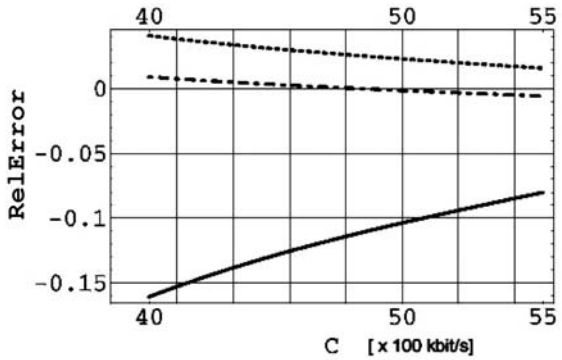


Figure 7.
The relative error of $\tilde{C}^{B\text{-}R}_{equ,sat}$, M2

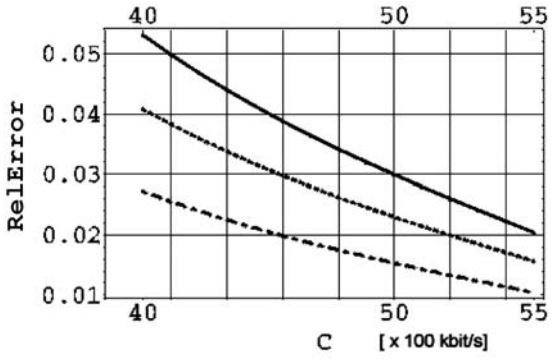


Figure 8.
The relative error of $\tilde{C}^{B\text{-}R}_{equ,WLR}$, M1

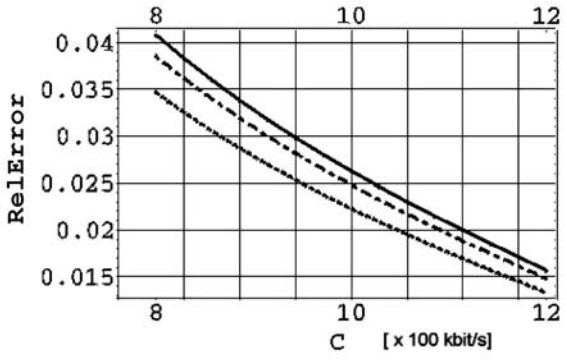


Figure 9.
The relative error of $\tilde{C}^{B\text{-}R}_{equ,WLR}$, M2

plotted with continuous line, while the dotted and dash-dotted lines refer to the equivalent capacity estimations for which $\tilde{\Lambda}_{X,ih}(s)$ or $\tilde{\Lambda}_{X,so}(s)$ was used.

The accuracy of the $\tilde{\Lambda}_{X,hoe}(s)$ based approximation is the worst in almost all cases, while the most accurate estimation is usually given by the one which uses the stochastic ordering based LMGF bound.

It can also be observed that the differences between the relative errors are bigger for smaller C values (i.e. for smaller γ values). It can also be seen that the investigated formulae give almost certainly a higher value than the exact one, if the $\tilde{\Lambda}_{X,ih}(s)$ or $\tilde{\Lambda}_{X,so}(s)$ bounds are used in the equivalent capacity estimator formulae. The absolute values of the relative errors decrease as γ increases (i.e. as the prescribed QoS level becomes more stringent).

6. Conclusions

In this article novel resource requirement estimator techniques were presented. With these methods the minimal required transmission capacity that should be provided for an aggregated network traffic flow in order to maintain a predefined QoS level can be computed. The most important advantage of our new formulae may be that they require very few input parameters: only the number of flows, their peak admission rates and the mean admission rate of the aggregated flow have to be known a priori.

For the computation of the presented equivalent capacity estimators the moment generating function of the aggregated traffic's rate distribution is needed. As it can not be determined exactly from the given parameters, three techniques capable to obtain an upper bound for the moment generating function was presented in Section 3. While the required parameters for these methods are the same, the performances of these bounds differ as we saw in Section 5.

Our numerical investigations also showed that for the best accuracy usually the new, stochastic ordering based bound should be applied in the equivalent capacity estimator formulae. However, if the computational simplicity is the most important factor, using the well-known Hoeffding bound may be the best idea.

With the aid of the presented resource requirement estimators efficient traffic load control mechanisms can be realized in packet based networks. The overload protection enables network operators to provide QoS guarantees for premium services, which in return ensures the satisfaction of their subscribers and encourages the evolution and spread of value-added services.

References

[1] C. Bouchat, S. van den Bosch, T. Pollet, "QoS in DSL Access", IEEE Communications Magazine, Vol. 41., Nr.9, pp.108–114, November 2003.

[2] W. Hoeffding, "Probability Inequalities for Sums of Bounded Random Variables", Journal of the American Statistical Association, 58:13–30, March 1963.

[3] Z. Heszberger, J. Zátonyi, J. Bíró, "Efficient Chernoff-based Resource Assessment Techniques in Multi-service Networks", Telecommunication Systems, 20(1):59–80, 2002.

[4] G. Mao, D. Habibi, "Loss Performance Analysis for Heterogeneous On-Off Sources with Application to Connection Admission Control", IEEE/ACM Transactions on Networking, 10(1):125–138, 2002.

[5] R. R. Bahadur, R. Rao, "On Deviations of the Sample Mean", Ann. Math. Statis., 31(27):1015–1027, 1960

[6] J. Y. Hui, "Resource Allocation for Broadband Networks", IEEE Journal on Selected Areas in Communications, 6(9):1598–1608, December 1988.

[7] M. Montgomery, G. de Veciana, "On the Relevance of Time Scales in Performance Oriented Traffic Characterizations", Proc. of the Conf. on Computer Communications, San Francisco, Vol. 2, pp.513–520, March 1996.

[8] J. Bíró, Z. Heszberger, F. Németh, M. Martinecz, "Bandwidth Requirement Estimators for Quality of Service Packet Networks", Proc. of the Intern. Network Optimization Conference, Evry, Paris, pp.95–100, October 2003.

Self-adaptive Multimodal User Interfaces based on Interface-Device Binding

SRIHATHAI PRAMMANEE, DR. KLAUS MOESSNER, PROF. RAHIM TAFAZOLLI

Mobile Communication Research Group, Centre for Communication System Research (CCSR)
University of Surrey, Guildford, UK

{S.Prammanee, K.Moessner, R.Tafazolli}@surrey.ac.uk

Key words: multi interface-device binding (MID-B), adaptive user interface, multimodal architecture

This paper introduces a mechanism that facilitates the dynamic shaping of human-machine interfaces in mobile environments. The necessity for self-adaptability of user interfaces for human-computer interaction (HCI) becomes increasingly important. Such adaptability provides the capacity to facilitate customised interaction depending on the system (and user) context based on automatic configuration of the user interface. The approach and work presented in this paper introduces a support architecture for self-adaptive user interfaces, based on distributed networked (interface) devices in a mobile communication scenario. The framework developed at the core of this work is called "Multi Interface-Devices Binding" (MID-B), it extends the concept of multimodality into the mobile environment, by allowing ad-hoc adaptation of available interface devices, (e.g. display, sound system, etc.). The here described structural design focuses on the definition of the basic architecture and of an extension to a service discovery protocol facilitating the dynamic real time binding of interface devices.

1. Introduction

Human-computer interaction (HCI) is concerned with the implementation of interactive computing systems for human use [1]. With the evolution of HCI [2], the focus extended from the concept of "interface" into "interaction revision" [3]. While "interface" means the sets of methods that can be called upon and used by a service, "interaction revision" refers to the procedure by which technology fits in a wide range of human needs, rather than what the interface looks like. HCI [4,5] introduces also the concept of "multimodal user interfaces", the basic idea of which is motivated by the knowledge of natural human communication, which makes use of the different modalities available.

The future use of multimodal user interface based services will be driven by 1) mobile and wireless technologies, aiming to provide any-time and any-where services, and 2) ubiquitous and pervasive computing, allowing applications to be jointly operated through multi interface devices located inside the service area.

User and Terminal mobility form a dynamic environment in which interface devices may appear and disappear depending on the facilities and (device) availability within the environment. To customise such service and to resolve an arbitrary interface change, raises a number of difficulties 1) to maintain a level of quality of the service in a random interface change scenario, 2) to provide the service that satisfies user preferences, 3) to support real time adaptive interface service and 4) to transform data content from a current modality to another modality offered by a different device [6,7].

These problems are caused by 1) the limited capability of small portable devices, 2) the characteristics of mobile networks and 3) the dynamic appearance and removal of interface modalities.

2. System Analysis

Multi Interface-Devices Binding (MID-B) system is the design that solves the above-mentioned difficulties and it is the general means of achieving interface self-adaptation in a mobile environment and distributed network by monitoring and estimating dynamic interface-devices that affect system deployment. Self-adaptation is the means by which internal system behaviour acclimatizes itself to either of the following:

- The effects of mobility on user interfaces may lead to changes in user (network) environment while maintaining the current interface configuration. The system needs to buffer the incoming data stream and alter internal actions of existing application by amending the implementation of data content delivered to optimal interfaces. For example, an application is running on a smart phone while a user is roaming from a private area to noise and public environment.
- The second mobility related effect would be that, due to modality accessibility to user interface, devices may change or connections between devices become corrupted. A re-device-allocation scheme will be developed to clarify the instructive changes in the system and maintain its service. The scheme will re-allocate modalities to the new physical resources and re-manage the system behaviour.

To facilitate the flexible binding and use of different modalities (user interfaces) in mobile environments, a number of system assumptions and definitions have to be made:

- *Distributed User Interface Devices* – are devices that provide at least one modality that can sense a human action or activate a human sense. The interface devices need to be equipped with a possibility to connect wirelessly.

- *Context Awareness* –
  depending on contextual knowledge, the choice of interfaces will be made. To simplify this part, the contextual knowledge for this system is restricted to the knowledge about availability of physical/distributed user interface devices and their capabilities.
- *Synchronisation of multiple interfaces* –
  is required to facilitate the associated step of modality fusion, for the here reported work, this part is limited to the provision of synchronisation information, the fusion functionality and mechanism is not described.
- *Content/User Data* –
  refers to the actual payload delivered via the user interfaces.

3. Multi Interface-Devices Binding (MID-B)

The *Multi Interface-Device Binding (MID-B system)* has the primary purpose of providing multimodal service via multiple interface devices in adaptive mobile and distributed networks. MID-B promises the self-adaptive interface binding by firstly allowing the system to be aware of arbitrary and capricious change of distributed interface devices and subsequently providing effective, flexible solution in responding to the change. There are a number of different possible approaches to design the MID-B prototype.

The prototype implementation system described here focuses on lightweight mobile terminals, using the third recommended approach by Niklfeld. Details stated in [8] (i.e. the communication model among different system modules). For the reason of providing profound service on non-powerful terminals, MID-B applies client/server-based architecture [9] and introduces vigorous network-end-point, Multimodal Service Base (MSB) being responsible for executing and maintaining satisfied multimodal interface services among distributed interfaces.

3.1. The components of MID-B

MID-B is made up of 3 primary components: Multimodal Service Base (MSB), a User Equipment Core-Device (UE-C) and one or more User Equipment Interface-Devices (UE-Is), see *Figure 1*.

- The *Multimodal Service Base (MSB)*
  acts as a network internal (multimodal) communication endpoint, consisting of a number of proxies and directories hosting proxy as well as device capability information.
- The *User Equipment Core-Device (UE-C)*
  refers to the portal (linking) device carried by users. There is only one UE-C active at any time that acts as the gateway between UE-I and MSB.
- The *Equipment Interface-Devices (UE-Is)*
  are a number of accessing and content rendering interface devices, for example microphone, screen, speaker etc.

These three components form the system and shape the architecture framework which functions are based on two assumptions:

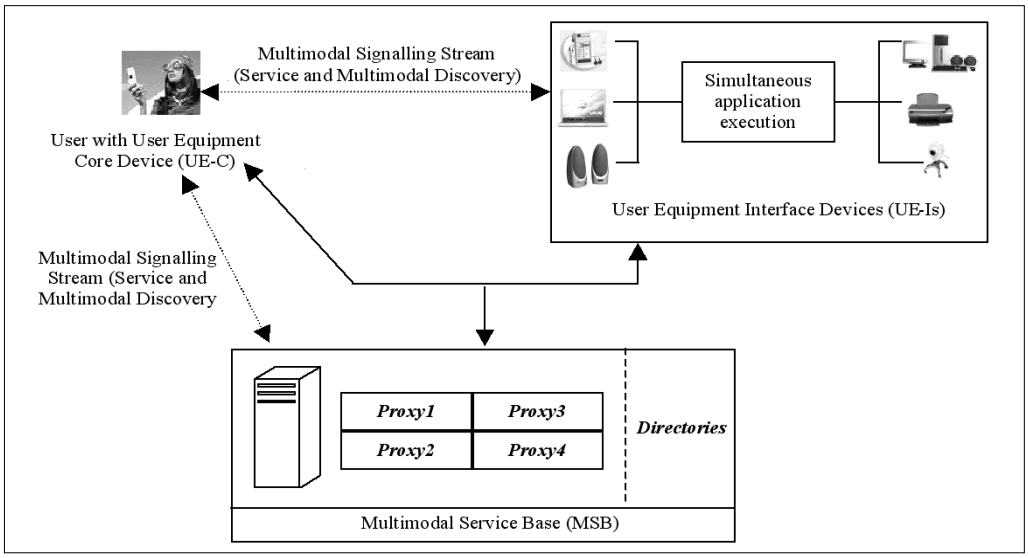
1) a user always carries a “portal device”, i.e. the User Equipment Core Device – UE-C, with the purpose of location detection. The system is not restricted to only a paired interface device, a number of UE-Is in the vicinity simultaneously perform the interface service, and

2) every interface node encloses wireless short range connectivity and discoverability (i.e. Bluetooth SDP). However, they have freedom of underlying network connectivity, such as WAN, LAN, Wireless LAN or cellular network.

3.2. The MID-B Mechanism

The MID-B mechanism describes the logical procedure of the overall system functions, see *Figure 2* for the class diagram. The UE-C acts as master, discovers slave devices (UE-Is) and requests their interface capabilities and modality service information using an extension to the Bluetooth SDP PDUs as message bearer. All available interfaces and UE-Is’ profiles including modalities, capabilities and connectivity are acknowledged and communicated back to the UE-C which consequently forwards all acknowledged services and modality information including its profile and data stream description to the MSB.

Figure 1. MID-B System (Physical Domain)



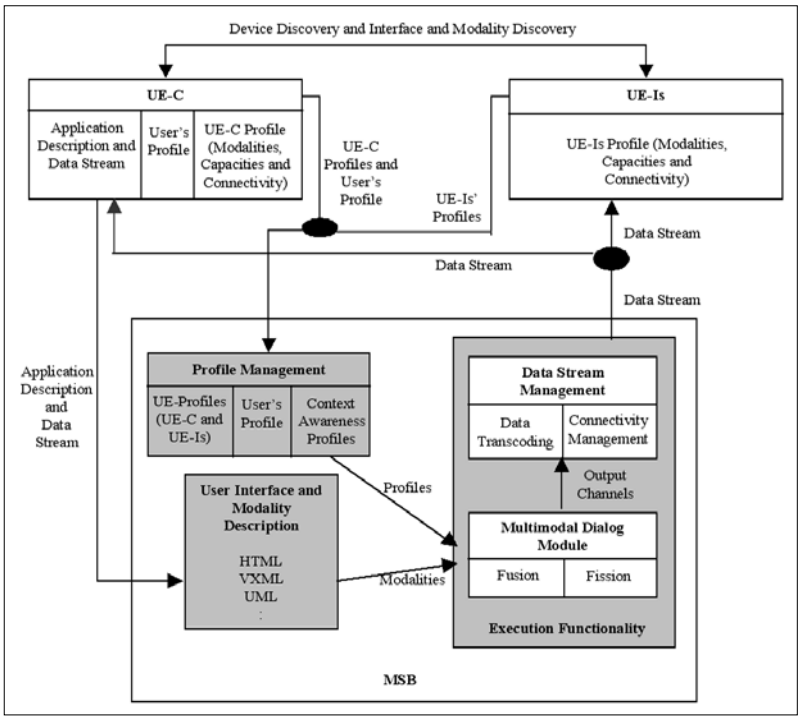


Figure 2. MID-B Class Diagram

Any discovered physical interface device at a time is considered as a candidate UE-I giving modality service. From the UE-I point of view, any interface is described by a name, supporting device, a list of attributes like connectivity and user privilege information. For example, a monitor is a visual output device providing a resolution of maximum 1024 by 168 pixels at 85 Hz frame rate, connected with fixed wire, and not bound in an adaptive user interface for a blind person, for another instance, a printer gives printing service which is able to offer colour printing at 120 by 720 dpi, and only stuffs in A building are able to use this printer. The MID-B algorithm has the task to establish what other modalities the system may need to support a running application (executed on the UE-C) and how such distributed interface components can be bound into the overall multimodal interface, rather than only searching where to find different physical devices.

To enable such mechanism, the UE-C and MSB maintain a mapping between UE-I's modality services and the network structure. The mapping result is stored in a proxy mechanisms inside the MID-B, this proxy provides

1) current network structure and available UE-Is,
2) information about how to invoke a particular user interface service offered by a UE-I.

The MSB includes directories which maintain all information about service proxies. Inside these proxies, there are three main sections designed for the current prototype; profile management, user interface and modality description as well as execution functionality.

- The profile management is composed of 1) user equipment profiles; including modalities, capacity and connectivity of UE-C and UE-Is, 2) context awareness profile, and 3) user profiles, sent from UE-C. In case a user has registered with the MID-B architecture beforehand, the system recognizes him/her and employs his/her profile as one of the factors to select appropriate/preferred input/output modality.
- User Interface and Modality Description maintains detailed accessing and rendering mechanisms, such as HTML for visual modalities, VoiceXML for audio modality and UML for a set of models that capture the functional and structural semantics of any complex information system.
- The execution function integrates the central algorithms that implement the management of the modalities; it combines and interprets all profiles and interface and modality description in order to make a decision about the most appropriate input/output modalities.

Additional mechanisms are included in the *Multimodal Dialog Module* which provides the information about the means by which each input modality is recognized and their information stream can be captured in a uni-modal stream, which is then delivered to a modality fusion module that is responsible for arbitration and interpretation. From system to user, a modality fission module decides the breaking and distribution of semantic information into parallel output channels (modalities).

The technology behind this amalgamates other well-established standard interface implementations, such as automatic speech recognition (ASR), graphical user interface (GUI) and natural language understanding [10,11]. After the selection of modalities has been completed (and the suitable interfaces are bound into the dynamic multi modal user interface), the application's content stream will subsequently be transcoded, transmitted and managed by the 'Data Stream Management'. 'Connectivity Management' controls the transport media and method of transmission, such as WLAN, Bluetooth or even Ethernet.

3.2. The MID-B Finite State Machine

The service provided by MID-B encompasses the provision of a set of user interface devices (UE-Is) that can be adapted to the current locally available set of interfaces. The prototype described implements a finite state machine modelling the connection and interaction between UE-C and UE-I, see *Figure 3.* Initially, the system is in idle state with an inactive UE-C and inactive UE-Is. Once the UE-C attaches to a service, the system initiates, by announcing itself, a discovery process. The UE-C then receives from any available UE-I a list of attributes or features. The UE-C then verifies whether the discovered UE-I can be used for the service session. After acknowledgement, the system enters the state 'Established' and UE-C can initiate the actual binding of the UE-Is, once this is completed, the user is

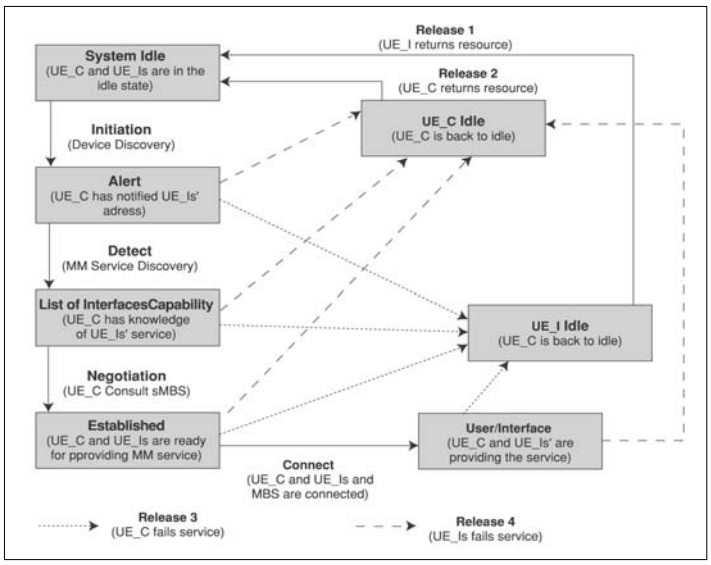


Figure 3. The 'MID-B' Finite State Machine

able to use the service through the chosen (and connected) modalities and interfaces. At any point, if a fault occurs, the state machine can fall back to the idle state (thus reverting to it's original configuration).

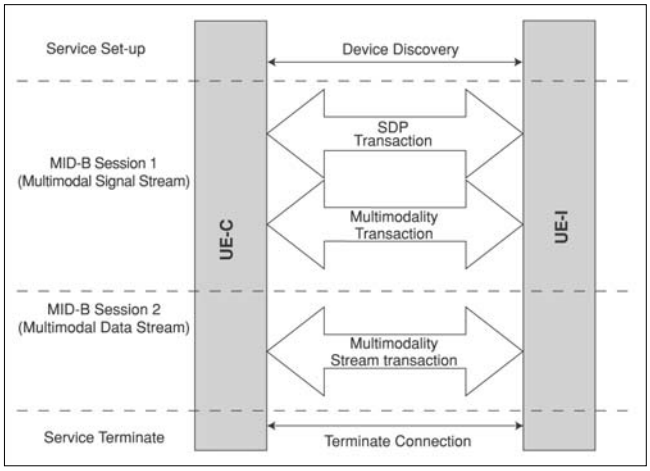
4. Multimodal Service Session on MID-B

The MID-B system processes two different styles of binding phases and streams, see *Figure 4*.

• The interface discovery and binding phase, transmitting "multimodal signaling stream" (maintaining and implementing discovery, negotiation and binding of the interface device). In addition to the function of the general Bluetooth service discovery protocol (SDP) [12], the interface discovery and binding phase operates new invention of multimodal discovery protocol for ascertaining knowledge of modality and device connectivity. Signal stream is the output of combination of SDP and multimodal discovery.

• The data transmission phase, transmitting "data stream", is actual user content for an application from/ to a connected interface device. With the fact that non-stricted device connectivity, for instance the media stream (i.e. video) may be re-routed to public display via wired network that offers more real time features.

Figure 4. Stages in Setting up a Multimodality Session



5. Conclusion

The work presented in this paper introduces the basic engines to manage the adaptation processes and initially describes their roles within the Multi Interface-Device Binding system. MID-B implements interface mapping by allowing a portal device (UE-C) to map external user interface devices in a temporary manner into an ad-hoc multimodal interface. Interface devices within the local area (UE-Is), can be bound into the overall multimodal interface in a dynamic manner.

The principles have been implemented in a prototype model based on a finite state machine. The model supports discovery, binding and release of interface devices at anytime, furthermore, in this work, an extension to the Bluetooth SPD PDU to facilitate interface device capability negotiation and binding is proposed.

References

[1] http://sigchi.org/cdg/cdg2.html#2_1.
[2] www.stanford.edu/class/cs147/lectures/04-history.html
[3] J.Karat, and C. M. Karat, "The Evolution of User-Centered Focus in the Human-Computer Interaction Field", IBM Systems Journal 42(4), Oct. 2003, pp.532–541.
[4] C. Duvallet, H. Boukachour and A. Cardon, "Intelligent and Self-Adaptive Interface", In Proc. of IEA/AIE'2000, LNCS 1821, Springer Verlag, New Orleans, US, pp.711–716.
[5] www.awprofessional.com/articles/article.asp?p=24103
[6] N. Becker, "Multimodal Interface for Mobile Client", Technical report TRITA-NA-E01102, Royal Inst. of Technology, Stockholm, Dec. 2001.
[7] A. Bailery, "Challenges and Opportunities for Interaction on Mobile Devices", Robust and Adaptive Information Processing for Mobile Speech Interfaces, Aug. 2004, Coling 2004 Satellite Workshop, Geneva, pp.9–14.
[8] G. Niklfeld and M. Pucher, "Mobile Multimodal Data Service for GPRS Phones and Beyond", Proc. of the Gourth IEEE International Conference on Multimodal Interfaces (ICMI'02), 2002.
[9] S. H.Meas, "Multi-Modal Browser Architecture: Overview on the Support of Multi-Model Browser in 3GPP", 2002.
[10] M. Haage, S. Schotz and P. Nugues, "A prototype Robot Speech Interface with Multimodal Feedback". In Proc. of IEEE RPMAN 2002, Berlin, 2002.
[11] www.w3.org/TR/emma.
[12] Bluetooth Specification Version 1.2 [Vol 3], 2003., pp.113–170.

Provable security for ad hoc routing protocols

GERGELY ÁCS, LEVENTE BUTTYÁN, ISTVÁN VAJDA

Budapest University of Technology and Economics, Department of Telecommunications
Laboratory of Cryptography and Systems Security (CrySyS)

{acs, buttyan, vajda}@crysys.hu

Reviewed

Key words: ad hoc networks, on-demand ad hoc source routing, secure ad hoc routing, provable security, simulation paradigm

In this article we present a new formal framework that can be used for analyzing the security of on-demand source routing protocols proposed for wireless mobile ad hoc networks. Our approach is based on the simulation paradigm which is a well-known and general procedure to prove the security of cryptographic protocols. We give the formal definition of secure ad hoc routing in a precise and rigorous manner using the concept of statistical indistinguishability. We present an ad hoc source routing protocol, called endairA, and we illustrate the usage of our approach by proving that this protocol is secure in our model.

1. Introduction

An ad hoc network is the cooperative engagement of a collection of wireless mobile nodes without the required intervention of any centralized access point or existing infrastructure. The nodes have terminal and network functions as well. They are often equiped by constrained energy supply (battery). Due to this fact and to reduce the interference of radio communication, the nodes communicate in a multi hop manner. In addition, due to the lack of a pre-deployed infrastructure, all nodes must perform routing and maintenance functions as well.

There exist two sorts of ad hoc routing: pro-active and reactive (or on-demand) protocols. In the rest of the article we deal with the latter one. Considering reactive routing, the source node initiates a route discovery towards a target node only if it needs to communicate with the target. In that case, the initiator node floods the whole network with route request messages *(rreq)*. Every node receiving the request appends its own indentifier to the node list that is placed in the request message and re-broadcasts the message. When the target node receives the route request it replies with one or more route reply messages *(rrep)* that contain the node list received in the request message. This node list itself is the discovered route. The reply travels back to the source node on the reverse of the route carried by the request.

Secure ad hoc routing means that the correct operation of the above mechanism is ensured even in the presence of an adversary. This has primary importance, since by manipulating the route discovery process, an adversary can paralyse the entire network using relatively small amount of resources.

Several “secure” ad hoc routing protocol have been proposed so far ([3] gives a deep overview of this topic), however the authors of these protocols have not proved their proposals by formal means. To the best of our knowledge, [2] is the first work that contains a precise mathematical model that is applicable for analyzing the security of ad hoc routing protocols. In [2], the authors present new subtle attacks against two well-known protocols (SRP, Ariadne) and at the same time they propose a new routing protocol that is provably secure in that model. The model that was used in [2] is based on the simulation paradigm which is a well-known method to give a formal proof of the security of various cryptographic protocols [4,5]. However, that model assumes a constrained Active-1-1 adversary that controls only one compromised device and uses only one compromised identifier. A further restriction is that the adversary can attack the execution of only one route discovery process.

In the present article, we generalize the model used in [2] for the case of an Active-y-x adversary and parallel execution of several insatnces of the routing protocol. Further, we show that endairA is secure in this extended model too. Due to space limitations, here we can only describe the basics of our approach; one can read about the complete work in [1].

2. Formal model

2.1. Modeling the network

We consider static ad hoc networks that are modeled by undirected labeled graphs $G(E,V)$, where each vertex uniquely corresponds to a node and there is an edge between two vertices if and only if the corresponding nodes can overhear each others communication (i.e., they are neighboring nodes). We assume that each node has an identifier that identifies the node unambiguously (e.g. a public key if we use public key cryptography). We further assume that all of the identifiers are authenticated but some of them are compromised by an adversary and the keys that are needed for their authentication are possessed by the adversary. An ad hoc network together with an adversary is shortly called a *configuration*. Formally a configuration is a triplet $G(E,V)$, $V^*, L)$, where $G(E,V)$ is a graph representing the network, $V^* \subset V$ is the set of nodes that are controlled by

the adversary, and *L* is a labelling function that assigns to each vertex the set of identifiers that belong to the node corresponding to this vertex. This set of identifiers is a singleton in case of honest nodes, but all of the corrupted identifiers are assigned to every corrupted node.

2.2. Modeling the adversary

We make the following assumptions about the adversary:

- the advesary cannot be physically present everywhere at the same time, thus it is not able to control the entire network;
- the adversay controls *x* nodes and uses *y* compromised identifiers (Active-*y*-*x* adversary, where $x, y \geq 1$);
- the set of compromised identifiers used by the adversary and the set of identifiers used by the honest nodes are disjoint;
- a corrupted node has the same communication capabilities as the honest nodes meaning that each malicous node can send messages only to its neighbors and they can overhear only the communication of neighboring nodes.
- the adversary is active in the sense of that besides eavesdropping messages, it can fabricate and insert new messages, and in addition it can modify and delay existing messages;
- the adversary is not adaptive meaning that it cannot coerce honest nodes to start route discoveries based on the information that it obtained in previously initiated route discoveries.

We assume that the initiator and the target of every route discovery process are honest.

2.3. Definition of plausible route

It is not a trivial task to give a formal definition of secure routing. The requirement of returning the most optimal (in some cases shortest) path seems to be a simple solution to this problem, however due to the varying message delaying and the applied optimizations, it seems to be an unrealistic requirement [2]. Further one can see that we cannot prevent a corrupted node from inserting arbitrary corrupted identifiers (even many times) into the node list carried by an intercepted routing message, and similarly we are not able to prevent the neighboring corrupted nodes from exchanging any information freely [1]. Now it should be clear that there are some attacks in practice which are unavoidable or it is very costly to defend against. Consequently, we have to form the notion of security with care: if we give a too strong definiton, then due to the unavoidable attacks mentioned above, no routing protocol will satisfy our definiton, and on the other hand if our definiton is too weak, then there will be protocols that are secure in our model but could be vulnerable to various attacks besides the unavoidable ones.

We solve this problem by embedding the possibility of unavoidable attacks in the definition of "correct routes". We call the routes that satisfy our definition of correctness *plausible* routes. The formal definition is given below. Every configuration can be unambiguously reduced to another configuration that has a graph without neighboring corrupted nodes. In other words we merge the neighboring vertices that correspond to neighboring corrupted nodes into a single one in the reduced graph. We denote this graph by $\underline{G}$.

Definition:

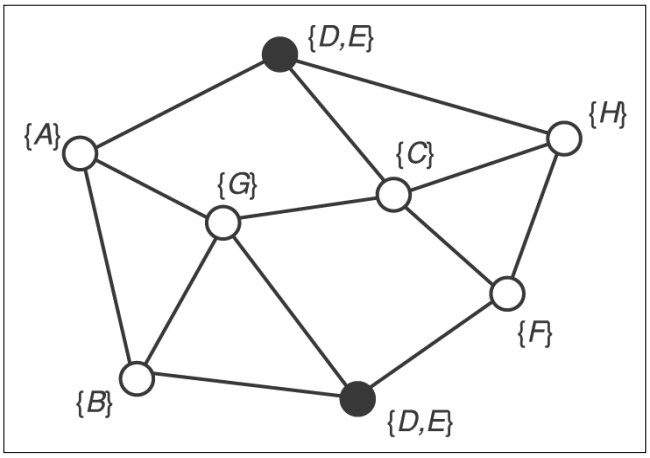
A sequence of identifiers is a plausible route, if

– it does not contain any repeating identifiers and

– it can be partitioned into sub-sequences in such a way that each of the resulting partitions is a subset of the identifiers assigned to a vertex in $\underline{G}$, and in addition, these vertices form a path in $\underline{G}$.

A reduced configuration is depicted on *Figure 1.* for illustration purposes. The solid verticies are representing the merged corrupted nodes that use compromised identifiers *D* and *E*. It is easy to see that *A,D,E,C,F* is a plausible route and a correct partitioning of this route is *A|D,E|C|F*, but *A,B,D,E,H* is not a plausible route, since the nodes that correspond to identifiers *E* and *H* are not neighboring.

Figure 1.



2.4. Simulation paradigm

The pivot of a formal model is to precisely define what we mean by secure routing. To achieve this goal we would like to apply the widely used simulation paradigm [4,5].

The fundamental idea of the simulation paradigm is that the adversary gains nothing if whatever it can achieve by unconstrained adversarial behaviour can also be achieved within essentially the same computational effort by a benign behaviour. The definition of the benign behaviour captures what we want to achieve in terms of security, and in our case, it is realted to the concept of plausible routes. In this model, we define a real-world model, that describes the real operation of the protocol under investigation, and the ideal-world one represents the ideal operation of this protocol. One can think of the real-world model as an implementation of the protocol, while the ideal-world model can be considered as a specification. Both models contain adversaries. The ideal-world adversary represents the

unavoidable attacks, or in other words, the tolerable imperfections of the system. On the other hand, we do not constrain the real-world adversary, but we assume that it can perform any polynomial-time attacks in the security parameter and in the size of the network.

A protocol is secure if for any unconstrained real-world adversary *A* there exist an ideal-world adversary *A'* such that *A* gains nothing more substantial than *A'* using the same computational effort. In other words, the behaviour of *A* can be simulated by the behaviour of *A'*, in the sense that the outputs of the ideal- and real-world models are indistinguishable from the point of view of the honest protocol participants. Intuitively, if any real-world adversary can be simulated by an appropriate ideal-world adversary, then there is no real-world adversary that can perform more than the unavoidable attacks.

In the followings we formally define the ideal- and real-world models of ad hoc routing protocols then we more precisely define the indistinguishability of the two models.

2.5. Real-world model

The real-world model is depicted on *Figure 2.* This model consists of the set of interacting and probabilistic Turing machines that communicate via common tapes. The machines model the operation of the honest protocol participants and the adversary. $M_1,...,M_{n-k}$ represent the honest devices that belong to honest nodes so they correspond to vertices v_i in $V \setminus V^*$.

The corrupted devices are denoted by $A_1,...,A_k$. These corrupted devices belong to corrupted nodes in V^*. Machine *C* models the radio links represented by the edges of $\underline{G}$. The task of machine *C* is to move the protocol messages appearing on the output tapes of the machines to the input tapes of the neighboring machines (neighboring relation is based on $\underline{G}$). Every machine apart from *H* is probabilistic. *H* is an abstraction of higher-layer protocols run by the honest parties meaning that *H* initiates the route discovery procedures by placing request messages on tape req_i. A response to these requests is eventually returned via tape res_i.

Tapes ext_j model an out-of-band channel through which the adversary can instruct the honest parties to initiate route discovery processes from an arbitrary node towards an arbitrary node. Arbitary in this context means that the adversary can choose these nodes. Note that the adversary is non-adaptive, thus it can use these tapes only at the beginning of the computation. So the messages placed on these tapes do not depend on the messages observed by the adversary during the protocol run.

At the beginning of the computation every machine is initialized with some input data (e.g cryptographic keys), which determines its initial state. The probabilistic machines also receive some random input (the coin flips to be used during the operation).When the machines have been initialized, the computation begins. The machines operate in a reactive manner, which means that they need to be activated in order to perform some computation. When a machine is activated, it reads the content of its input tapes, processes the received data, updates its internal state, writes some output on its output tapes, and goes back to sleep (i.e., starts to wait for the next activation). The machines are activated in rounds by a hypothetic scheduler in a specified order. The computation ends when *H* reaches one of its final states.

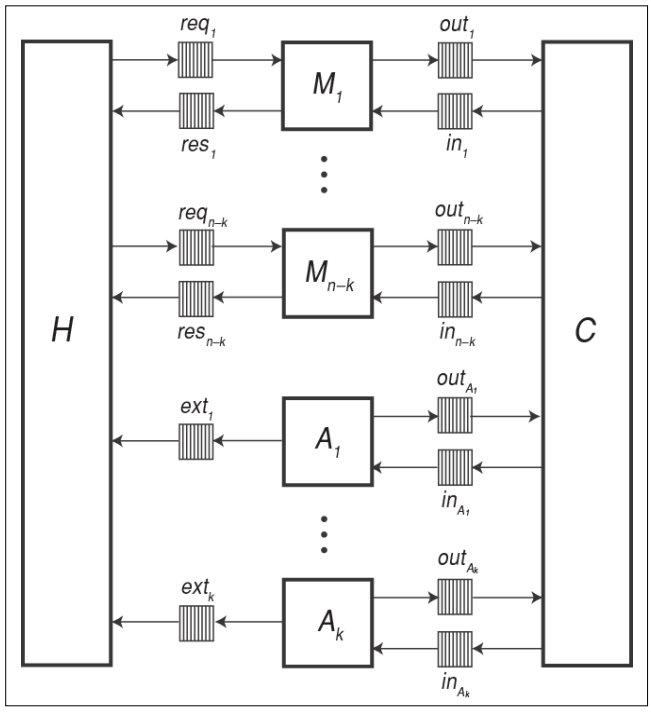
The output of the real-world model is the set of the routes returned to *H*. This output is denoted by $real\_out_{conf,A}(r)$, where *conf* and *A* represent the configuration and the adversary respectively. $r = (r_l, r_{M1}, ..., r_{Mn-k}, r_{A1}, ..., r_{Ak}, r_C)$ is a vector containing the random input of each machine and r_l is the random input used to generate the cryptographic keys. $real\_out_{conf,A}$ denotes the random variable describing the output, when *r* is chosen uniformly at random.

2.6. Ideal-world model

The ideal-world model is shown in *Figure 3.* As one can see, the construction of the ideal-world model is similar to the construction of the real-world model, so here we only describe the differences between them:

- Before machine *C'* places a route reply message *(rrep)* on the tape in_i of a machine M_i, it checks whether the message contains any non-plausible routes. If and only if this is the case, then *C'* puts a corruption flag on the message. Otherwise machine *C'* operates like machine *C*.
- When M_i' receives a route reply message that belongs to a request that was initiated by him, then he performs all of the verifications required by the protocol on the message. If these verifications are successful, then it checks whether the message has a corruption flag. If it has, then M_i' drops the message. Otherwise machine M_i' operates like machine M_i.

Figure 2. The real-world model



The output of the ideal-world model is the set of the routes returned to *H*. This output is denoted by *ideal_out*$_{conf,A}$(r), where the interpretation of *r'* is similar to the interpretation of *r* in the real-world model.

ideal_out$_{conf,A}$ denotes the random variable describing the output, when *r'* is chosen uniformly at random.

It is easy to see that, in the ideal-world model, *H* never receives a route reply message containing any non-plausible routes. In effect, the ideal-world model is ideal in that sense.

2.7. Formal definition of secure ad hoc routing

Considering the comments related to the unavoidable attacks, we require from a secure routing protocol to return non-plausible routes only with negligible probability. We can formally describe this requirement using the two models and the simulation paradigm in the following way:

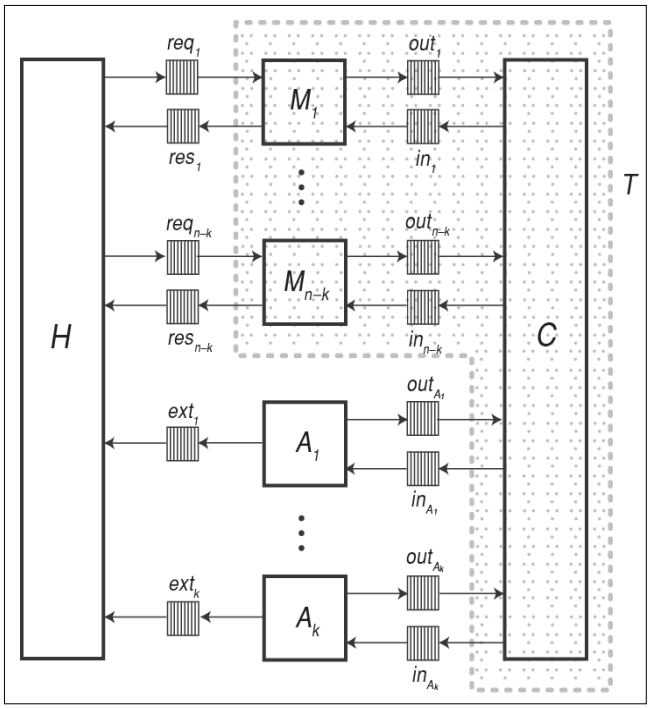
Definition:

A routing protocol is said to be statistically secure if, for any configuration conf and any real-world adversary A, there exists an ideal-world adversary A', such that ideal_out$_{conf,A}$ *is statistically indistinguishable from real_out*$_{conf,A'}$.

In this definition we do not require the exact matching of the distributions, since that requirement could not be satisfied by any protocol in practice. The adversary can always carry out a successful attack against the applied cryptographic primitve with negligible probability (e.g. by forging a correct digital signature).

The above definition can be weakened if we require computational indistinguishability instead of statistical indistinguishability, but in this article we will not need this.

Figure 3. The ideal-world model



3. Security of endairA

In this part we would like to demonstrate the usage of our model by a short example. First, we present a new ad hoc routing protocol, called endairA [2,1], and then we prove that this protocol is secure in our model.

The operation of the protocol is exemplified by the following message exchanges, where sig_x denotes the digital signature of node *x*, and *id* is a non-predictable random request identifier:

Route Request:
S→* : [rreq, S, D, *id*, ()]
B→* : [rreq, S, D, *id*, (B)]
C→* : [rreq, S, D, *id*, (B, C)]

Route Reply:
D→ C : [rrep, S, D, *id*, (B, C), (sig_D)]
C→ B : [rrep, S, D, *id*, (B, C), (sig_D, sig_C)]
B→ S : [rrep, S, D, *id*, (B, C), (sig_D, sig_C, sig_B)]

In endairA, the initiator of the route discovery process generates a route request *(rreq)*, which contains the identifiers of the initiator (**S**) and the target (**D**), and a randomly generated request identifier *(id)*. Each intermediate node that receives the request for the first time appends its identifier to the route accumulated so far in the request, and re-broadcasts the request. When the request arrives to the target, it generates a route reply *(rrep)*. The route reply contains the identifiers of the initiator and the target, the accumulated route obtained from the request, and a digital signature of the target on these elements. The reply is sent back to the initiator on the reverse of the route found in the request. Each intermediate node that receives the reply verifies that its identifier is in the node list carried by the reply, and that the preceding identifier (or that of the initiator if there is no preceding identifier in the node list) and the following identifier (or that of the target if there is no following identifier in the node list) belong to neighboring nodes. Each intermediate node also verifies that the digital signatures in the reply are valid and that they correspond to the following identifiers in the node list and to the target.

If these verifications fail, then the reply is dropped. Otherwise, it is signed by the intermediate node, and passed to the next node on the route (towards the initiator). When the initiator receives the route reply, it verifies if the first identifier in the route carried by the reply belongs to a neighbor. If so, then it verifies all the signatures in the reply. If all these verifications are successful, then the initiator accepts the route.

The proof of the following theorem illustrates how the framework introduced earlier can be used in practice.

Theorem:

endairA is statistically secure if the signature scheme is secure against chosen message attacks.

Sketch of the proof: A routing protocol is statistically secure if it returns a non-plausible route for any *conf* configuration and for any *A* adversary only with negli-

gible probability. More precisely we have to prove that a *rrep* message in the ideal-world model of the protocol is dropped due to its corruption flag only with negligible probability.

Let us suppose that following message is dropped due to its corruption flag in the ideal system, while the real system does not drop it:

$msg = [\text{rrep}, S, D, id, (N_1, N_2, \ldots, N_p), (sig_D, sig_{Np}, \ldots, sig_{N1})]$

In that case we can conclude the following:
- there is no repeating identifier in route $\pi = (S, N_1, N_2, \ldots, N_p, D)$;
- N_1 is a neighboring node of node S;
- every signature is correct;
- S and D are honest nodes;
- every intermediate node (with overwhelming probability) sees the route that was sent by node D (π), since node D signed that route, and every intermediate node checks this signature;
- despite all the above properties, π is a non-plausible route in graph $\underline{G}$, where $\underline{G}$ is the graph of the reduced configuration.

We prove that this can only be possible if adversary A has successfully forged the signature of at least one honest node. We know that there is no neighboring vertices in the graph of the reduced configuration that correspond to neighboring corrupted nodes in the network and in addition each non-corrupted node uses a single and unique non-compromised identifier. It follows that every route, including $(N_1, N_2, \ldots, N_p)$, has a unique meaningful partitioning, which is the following: each non-compromised identifier, as well as each sequence of consecutive compromised identifiers should form a partition.

Let $P_1, P_2, \ldots, P_k$ be the unique meaningful partitioning of the route $(N_1, N_2, \ldots, N_p)$. The fact that this route is non-plausible implies that at least one of the following two statements holds:

1. There exist two partitions $P_j = \{N_i\}$ and $P_{j+1} = \{N_{i+1}\}$ such that both N_i and N_{i+1} are non-compromised identifiers, and the corresponding non-corrupted nodes are not neighbors.
2. There exist three partitions $P_j = \{N_i\}$, $P_{j+1} = \{N_{i+1}, \ldots, N_{i+q}\}$, $P_{j+2} = \{N_{i+q+1}\}$ such that N_i and N_{i+q+1} are non-compromised and $N_{i+1}, \ldots, N_{i+q}$ are compromised identifiers, and the non-corrupted nodes that use N_i and N_{i+q+1} have no common corrupted neighbor.

In Case 1, N_{j+1} does not sign the route reply, since it is non-corrupted and it detects that the identifier that precedes its own identifer in the route does not belong to a neighbor. Hence, the adversary must have forged sig_{Ni+1} in *msg*.

In Case 2, the situation is more complicated. Let us assume that the adversary has not forged the signature of any of the non-corrupted nodes.

N_i must have received

$msg' = [\text{rrep}, S, D, id, (N_1, N_2, \ldots, N_p), (sig_D, sig_{Np}, .., sig_{Ni+1})]$

from a corrupted neighbor, say v^*, since N_{i+1} is compromised, and thus, a non-corrupted node would not send out a message with sig_{Ni+1}. In order to generate *msg'*, node v^* must have received

$msg'' = [\text{rrep}, S, D, id, (N_1, N_2, .., N_p), (sig_D, sig_{Np}, .., sig_{Ni+q+1})]$

because by assumption, the adversary has not forged the signature of N_{i+q+1}, which is non-compromised. Since v^* has no corrupted neighbor, it could have received *msg''* only from a non-corrupted node. However, the only non-corrupted node that would send out *msg''* is N_{i+q+1}. This would mean that v^* is a common corrupted neighbor of N_i and N_{i+q+1}, which contradicts the assumption of Case 2. This means that our original assumption cannot be true, and hence, the adversary must have forged the signature of a non-corrupted node.

Consequently, if a reply message like *msg* can occur in the ideal system with non-negligible probability then the adversary is able to forge the signature of a non-compromised node with non-negligible probability. It contradicts our assumption that the used signature scheme is secure.

4. Summary

In this article, we presented a formal model in which we defined in a precise and rigorous way what we mean by secure ad hoc routing. Using the proposed model, one can prove (or fail to do so) the security of on-demand source routing protocols. We demonstrated the practical usage of the model on a real example, namely, we proved that endairA is secure in our model. In the near future, we will define a similar model to analyze the security of on-demand distance vector routing protocols (e.g. ARAN, S-AODV). Further, we would like to automate the process of proofs by an adequate formal language (e.g. process algebra) and related verification tools.

Acknowledgment

The work presented in this paper has been supported by the Ministry of Education (BÖ2003/70), the Hungarian Scientific Research Fund (T046664) and IKMA.

References

[1] Ács G.: Ad hoc útvonalválasztó protokollok bizonyított biztonsága, TDK dolgozat, 2004. nov. 9.
[2] L. Buttyán, I. Vajda: Towards provable security for ad hoc routing protocols, In Proc. of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks, Washington D.C., USA, October 2004.
[3] Y.-C. Hu, A. Perrig: A survey of secure wireless ad hoc routing, IEEE Security and Privacy Magazine, June 2004.
[4] O. Goldreich: The foundations of modern cryptography, Cambridge University Press, 2001.
[5] M. Bellare, R. Canetti, H. Krawczyk, A modular approach to the design and analysis of authentication and key exchange protocols, 1998. In Proc. of the ACM Symp. the Theory of Computing

Fiber-delay lines for intensity noise suppression in optical links

MÁRK CSÖRNYEI, TIBOR BERCELI

Budapest University of Technology and Economics,
Department of Broadband Infocommunication and Electromagnetic Theory
{csornyei, berceli}@mht.bme.hu

Reviewed

Key words: semiconductor lasers, intensity noise, optical-microwave filtering, noise suppression, coherence

In case of short haul optical links, optical local area networks or optical-mobile networks the most important noise source is the relative intensity noise (RIN) of the laser diodes. This paper will report on a new all-optical technique of intensity noise suppression for semiconductor lasers. The new scheme we have used is based on an Unbalanced Mach-Zehnder Interferometer (UMZI), which is able to cancel the intensity noise enhancement in the microwave domain and thus improve the link signal-to-noise ratio. Extending the UMZI to fiber-delay line filter the noise reduction capability can be further increased. Additionally the condition of stable, incoherent operation is detailed.

1. Introduction

The increasing demand for new telecommunication applications and higher data rates requires continuous research concerning the technical parameters of the state-of-the art optical networks. In addition the optical signal processing solutions need further improvement in the characteristics of photonic devices as well.

One of these most important physical parameters of the optical link is the *Relative Intensity Noise (RIN)* of laser sources, which has major influence on the detected signal-to-noise ratio especially in short haul optical transmission.

The spectral density of the RIN is not equally flat around the optical carrier. It has a remarkable increment at the relaxation oscillation frequency which is defined by the laser internal operation. In case of laser diodes the relaxation oscillation frequency, in connection with the intensity noise maximum overlaps with the modulation information in the microwave band, which causes a quality degradation in the signal transmission. It is now obvious that in future high transmission capacity optical networks the intensity noise suppression is a crucial problem.

Before revising the various possible ways of RIN suppression it is worth to dealing with the causing effects of the intensity noise themselves. The main reasons are typically the temperature fluctuations, the spontaneous emission of the laser sources and the optical reflections caused by the refractive index changes at light coupling into fibers or other optical devices.

To cancel the intensity noise originating in optical reflections optical isolators can provide an appropriate solution. In this case the reflected signal parts are significantly attenuated and can not contribute to developing the intensity noise. Optical isolators present a right possibility for noise suppression but only in cases where the RIN is mostly generated by coupling reflections.

In order to optimize the signal-to-noise ratio there is a further opportunity in carefully adjusting the laser diode biasing. Increasing the bias current the relaxation oscillation of the laser diode shifts to higher frequencies with a smaller resonance amplitude, which means a decrease of the intensity noise maximum. Properly setting the bias current the noise enhancement can be moved out of the selected transmission band and suppressed. This way of noise cancellation is useful only in case of narrow band modulation. The further drawback of this solution goes back on the fixed value of the bias current, which is why the source output power cannot be adjusted freely and we lose on flexibility of the network. Adding or removing nodes of the optical network should influence the output laser power which is impossible if the biasing is kept constant to reduce the intensity noise.

For intensity fluctuation suppression of solid-state lasers the optical feedbacking of the laser crystal output power means a well known and efficient way [1,2]. With the design of an optoelectronic control loop remarkable suppression can be achieved at the frequency of the relaxation oscillations. This method can be very well used for the high peak of the low frequency (<10MHz), narrow band intensity noise of solid-state lasers, but in case of the high frequency (>1GHz), broadband noise increment of laser diodes it is unusable.

The *Unbalanced Mach-Zehnder Interferometer (UMZI)* presents a uniform solution for noise suppression both for laser diodes and solid-state lasers (Nd:YAG, Nd:YVO_4) [3]. This approach, which is dealt with in this paper, has more advantages than the ones mentioned so far. The UMZI based intensity noise cancellation exclusively utilizes passive optical devices therefore it sports all the advantages of optical signal processing, i.e. insusceptible to electromagnetic interferences (EMI), it does not need electrical biasing and in comparison with copper based electronic systems it can be realized in smaller sizes and from the more economical SiO_2.

The structure of the paper is as follows. Section 2 accounts for the fact that intensity noise suppression of different laser sources is unbearable. Section 3 illustrates in detail the operation of the unbalanced (asymmetric) Mach-Zehnder interferometer in an intensity noise suppression application, the measurement results and the comparison of coherent and incoherent working regime. Section 4 summarizes the results so far and further possible efforts in this field.

2. Noises of optical transmissions

In case of intensity modulated optical transmission using PIN photodetector the transmission noise comprises three terms, which are the shot noise, the receiver thermal noise and the relative intensity noise of the laser source [4]. Supposing independent noise sources, the following signal-to-noise ratio can be formulated at the output of the photodetector

$$\frac{S}{N} = \frac{I^2}{\sigma_s^2 + \sigma_t^2 + \sigma_R^2} \quad . \tag{1}$$

In the numerator of (1) there is the square of the photodetector current while in the denominator there is the variance of the noise source currents. The variance of the three noise components are expressed in (2-4):

$$\sigma_s^2 = 2eB(I_p + I_d) \tag{2}$$

$$\sigma_t^2 = 4k_B T_0 B / R_L \tag{3}$$

$$\sigma_R^2 = \frac{\eta^2 e^2}{(hf)^2}(RIN)P^2 B \tag{4}$$

In the formulas of (2-4) e stands for the charge of the electron, k_B for the Boltzmann-constant, B for the bandwidth of the photoreceiver, P for the optical power, η for the quantum efficiency, h for the Planc-constant, I_p for the photocurrent and I_d for the dark current. RIN gives the level of the relative intensity noise to the optical carrier. It is obvious that both the level of the shot noise and the intensity noise depend on the value of the incoming optical power, while the thermal noise is only dependent on the receiver temperature and its load resistor.

When substituting the typical parameters of the optical links of today operating in the 1550-nm band for the equations (2-4), the relative intensity noise can oversize the impact of the thermal noise based upon the -130, -150 dB/Hz RIN values typical of *Fabry-Perot (FP)* lasers. However, as the value of (4) decreases by increasing the transmission length and the link attenuation, over certain fiber lengths and network sizes the component deriving from intensity noise sinks bellow the thermal noise of the receiver, therefore its impact will be irrelevant.

It is clear that the RIN level and the noise suppression methods aiming at the amplitude fluctuation are most relevant in case of short haul optical transmissions, optical-mobile systems and optical *LMDS-s (Local Multipoint Distribution System)*. On the basis of calculations of [5] the relative intensity noise is the determining noise source of optical links up to about 30 km length.

From these facts it emerges that further improvement of the quality of local and urban optical network transmissions is only feasible by using RIN noise reduction methods.

3. Interferometer in noise suppression application

The Unbalanced Mach-Zehnder Interferometer based intensity noise suppression scheme for laser diodes is depicted in *Figure 1.*, the laser output is coupled into an Unbalanced Mach-Zehnder Interferometer. The input 3 dB coupler divides the laser signal into the two arms of the UMZI. Properly setting the time delay difference between the two signal paths the output 3 dB coupler combines the signals with a phase shift of 180° at the relaxation oscillations frequency. Exploiting this time delay difference, the intensity noise peak can be appreciably reduced.

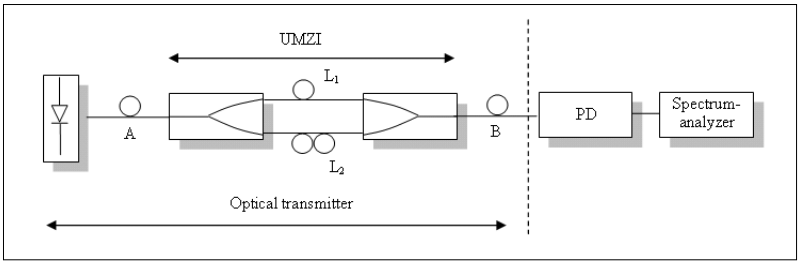
In that structure we have utilized an InGaAsP *Multi-Quantum Well (MQW)* Fabry-Perot (FP) laser diode. The output power was 0.1-2 mW. The pigtailed output of the laser diode was connected to the interferometer, which consisted of two Kamaxoptic 3 dB (50/50) splitter modules and two SMF-28 type single mode optical fiber in-between.

The intensity noise maximum defined by the relaxation oscillation is at 2 GHz exciting the diode by a bias current of 10 mA. According to (5) an UMZI path length difference of 0.05 m is required in order to reduce the noise at 2 GHz [6].

$$\tau = T_2 - T_1 = \frac{n}{c}(L_2 - L_1) = \frac{1}{FSR} \Rightarrow \Delta L = \frac{c}{n} \cdot 250\,ps = 5cm \tag{5}$$

In (5) n stands for the fiber effective refractive index, c is velocity of light in vacuum, L_1, L_2 and T_1, T_2 are the UMZI arm lengths and the delays respectively. Based on the delay difference the *Free Spectral Range (FSR)* of the interferometer can be calculated, which gives the frequency difference of the periodic suppressions in the transmission function. In the case of a 5 cm length difference the FSR is 4 GHz as it is depicted in *Figure 2.*

Figure 1.
Unbalanced Mach-Zehnder Interferometer for intensity noise suppression of laser diodes.



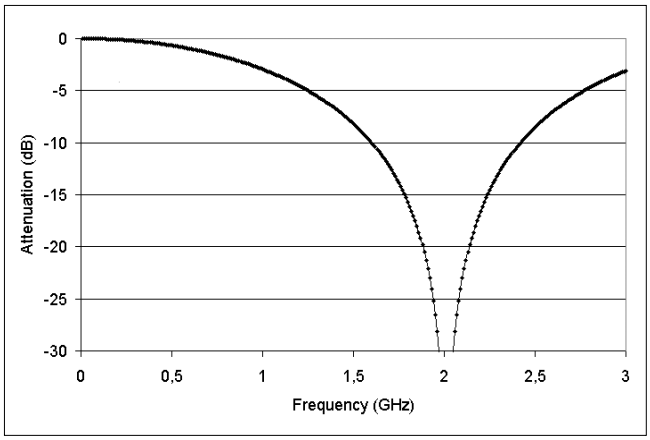


Figure 2. UMZI transfer function. (FSR = 4 GHz, ΔL= 5 cm)

Since the by the 5 cm fiber length difference defined 250 ps delay difference is much shorter than the coherence time of the laser source the proposed interferometer works in the strong coherent regime. The coherence time of the our FP laser diode is 3 ns which can be calculated by (6) from the laser spectral linewidth (Δν=100 MHz).

$$\tau_c = \frac{1}{\pi\Delta\nu} \qquad (6)$$

During coherent operation it is not the desired intensity-based summery that occurs at the interferometer output but the basis of the interference is the field intensity spreading in the fiber [7]. Formulating it in another way, while the interferometer in the incoherent working regime can be regarded as a linear network concerning optical intensity and the interference only influences the envelope realized by intensity modulation, in the coherent case the optical carrier can fall prey to interference [8].

In the coherent case, i.e. if the difference in the arm length of the interferometer acting as a filter is less than the coherence length of the laser ($\tau<\tau_c$), the transmission function becomes extremely sensitive and instable since the output signal appears or disappears accidentally because of the interference also touching the carrier. While owing to very little differences in the interferometer arm lengths it is wearisome to maintain appropriate operation in the coherent regime, which is only possible by constantly supervising the optical phase and the system temperature and adjusting the biasing of the laser source very accurately, in case of larger differences in the fiber lengths and less FSR the noise reduction has to be realized in the incoherent regime.

In order to ensure the incoherent operation a fiber interferometer with a Free Spectral Range of 200 MHz (path length difference (ΔL): 1 m, n=1.5) was chosen instead because of its longer delay differences and stable incoherent operation. The noise suppression feasible with this structure is shown in *Figure 3.* and *Figure 4.* show the measured transfer function of the interferometer discussed above and the achieved noise reduction respectively. The interferometer has an attenuation of about 6 dB which comes from the attenuation of optical connectors between the laser pigtail, the 3 dB couplers and the fibers. Taking account of this attenuation there is a noise reduction of 8-9 dB at the UMZI resonance frequencies around 2 GHz in Figure 4. The further suppression is possible at the selected frequencies but the measurement is limited due to the spectrum analyzer noise floor.

Actually the UMZI is an optical *FIR (Finite Impulse Response)* Filter which has got only two taps and both of the filter coefficients are +1. Since we only have positive values for the filter coefficients the UMZI behaves as an optically realized low-pass filter with multiple transmission and attenuation bands. The low-pass characteristic is of prime importance because it ensures that the optical carrier itself will not be filtered out. Using UMZI, noise reduction is only possible at selected resonance frequencies of the interferometer *(Figure 5)*.

Figure 3.
UMZI with a Free Spectral Range of 200 MHz.
The measurement was done between the points of A and B of the structure depicted in Fig.1.
The interferometer has an attenuation of 6 dB and a noise suppression capacity of 15 dB at the selected resonance frequencies.

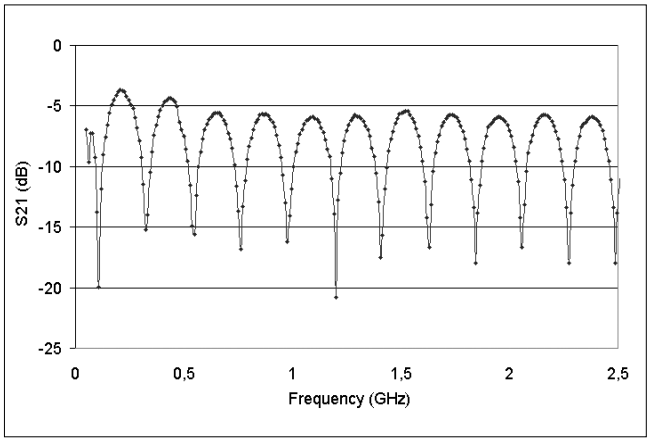
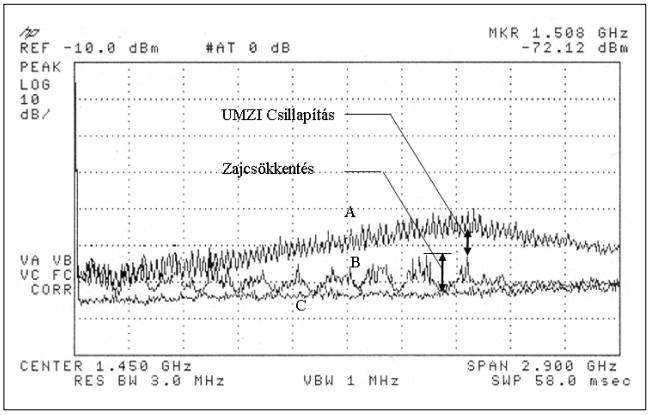


Figure 4.
Measured noise suppression of the UMZI structure of Fig.1.
A) Relative Intensity Noise of the investigated FP laser diode at 2 GHz.
B) Noise suppression realized by the interferometer.
C) Noise floor of the measurement setup. Measurement conditions: ResBW = 3 MHz, No Video Averaging, Input Attenuation = 0 dB.



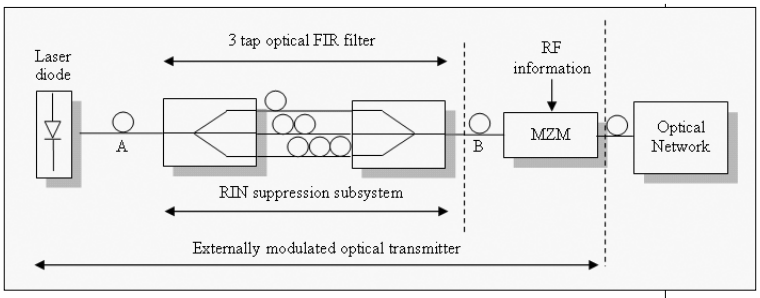


Figure 5.
3 tap optical transversal filter for laser diode noise cancellation. After the noise suppression blocks the information can be modulated with an external optical filter.

To achieve overall noise suppression around the relaxation oscillations of the laser diode, the interferometer should be extended with additional fiber arms. It means we should increase the tap number in our optical FIR filter. Placing new lines with different optical delays will result in spectral broadening of the attenuation bands in the filter transfer function. During the design of the optical transversal filter it is very important to fulfill the conditions of the incoherent operation, which means in our case that all the fiber length differences should be longer than the laser source coherence length.

Formulating the same in the frequency domain means, that in order to achieve a noise suppression at 2 GHz the interferometer with a FSR of 4 GHz should be replaced by an optical-microwave filter with a transfer function consisted of many narrower FSR-s. Taking into account these requirements concerning coherence we will end up with slightly different design methodology than in the case of commercial optical-microwave filters [8].

For calculation of the interferometer suppression frequencies (f_0) the well known [7] formula of (7) can be used.

$$f_0 = (2k+1)\frac{c}{2n_{eff}\Delta L} = \frac{2k+1}{2\Delta T} \qquad (7)$$

According to (7) for a RIN suppression at 2 GHz a FSR of 4 GHz is required which gives a delay difference of 250 ps (k=0). Intending to realize incoherent operation it is worth to setting the value of the constant k higher. Using k=11 and forcing f_0 to be 2 GHz a delay difference of 5750 ps will work out.

Using delay differences bigger than the 3 ns coherence time the transmission functions of *Figure 6.* and *Figure 7.* are feasible. Evaluating the structure of Figure 6. the noise suppression of *Figure 8.* can be achieved, where the noise cancellation band is pushed up to 400 MHz.

4. Conclusions

In our paper we have suggested passive all-optical solution for the suppression of relative intensity noise of local optical networks. We have looked into and presented the noise suppression that can be obtained by the asymmetric Mach-Zehnder interferometer. For widening the suppression band and securing stable incoherent operation we have introduced novel ideas that differ from the traditional filter design concepts.

Our results perfectly fit into the future concepts of fiber systems using only optical devices while lacking electronical signal processing elements.

Further objectives of our research are to analyze the impact of laser phase noise as well as to supervise the possibilities of the integrated optical implementation.

Acknowledgment

The authors acknowledge the Grant of the "National Research Foundation" (OTKA) No. T042557 and the European project Gandalf IST-1-507781-STP.

Figure 6.
Transfer function of the 3 tap incoherent optical transversal filter of Fig.5. The delay differences are: 3.25ns and 5ns. Coherence time: 3ns.

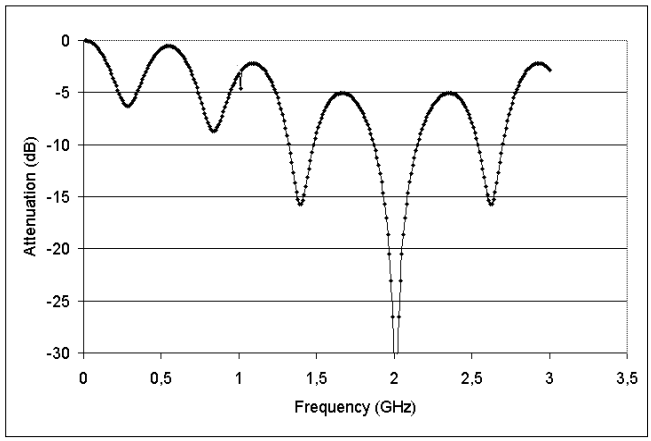
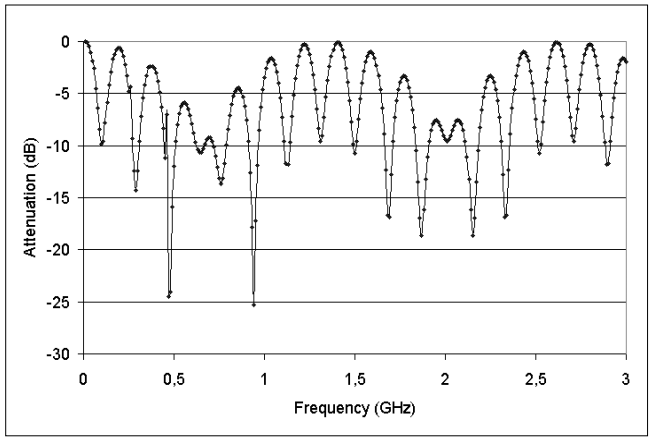


Figure 7.
Transfer function of the 3 tap incoherent optical transversal filter of Fig.5. The delay differences are: 5.75 ns and 5 ns. Coherence time: 3 ns.



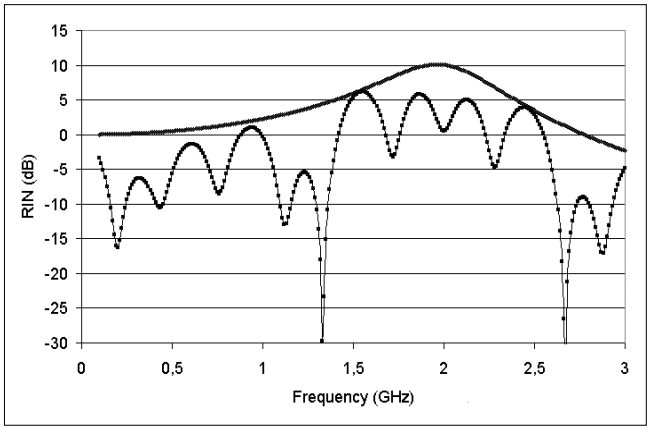


Figure 8.
Calculated noise suppression results based on the filter transfer function of Fig.6. Solid line: the calculated relative intensity noise of the investigated Fabry-Perot laser diode. There is a noise suppression of 10 dB in the range of the 2 GHz relaxation oscillation.

References

[1] T. J. Kane,
"Intensity noise in diode-pumped single-frequency Nd:YAG lasers and its control by electronic feedback", IEEE Photon. Techn. Letters, Vol. 2, No.4, 1990. április

[2] M. Csörnyei, T. Berceli, P. R. Herczfeld,
"Noise suppression of Nd:YVO4 solid-state lasers for telecommunication applications", J. Lightw. Techn., Vol. 21, No.12, pp.2983–2988. 2003. december

[3] M. Csörnyei, T. Berceli, T. Marozsák,
"All-optical intensity noise suppression of solid-state lasers for optical generation of microwaves", XV International Conference on Microwaves, Radar and Wireless Communications, MIKON-2004, Varsó, Lengyelország, pp.781–784. 2004. május

[4] Frigyes I.,
"Hírközlő rendszerek", Műegyetemi Kiadó, 1998.

[5] Marozsák T.
"Félvezető lézerek alkalmazása és modellezése segédvivős optikai rendszerekben", Doktori értekezés – BME, Budapest, 2004.

[6] B. Cabon, V. Girod, G. Maury,
"Optical generation of microwave functions", Proc. OMW2000 Summer School, Autrans, France, 2000. szeptember

[7] A. Hilt,
"Basics of microwave network analysis of optical circuits", Optical/Wireless Workshop in the framework of the European MOIKIT project, Budapest, 2001. március

[8] J. Capmany,
"Fiber-optic filters for RF signal processing", Proc. OMW2000 Summer School, Autrans, France, 2000. szeptember

News

The Enhanced Long Distance Services Solution works together with Veraz Switching family and provides revenue streams for both carrier networks at any point in their transition from TDM to IP. Based on Veraz's programmable service engine, the LD services are easily customizable by the carrier or by 3rd parties.

The following the services available:
- Account Codes
- Number Translations (Toll Free)
- Personal Toll Free
- Security Toll Free
- Unauthorized User Redirect
- Tariff Announcements
- Pre-Subscribed International Long Distance
- Automated Collect Call
- Hotel PBX Billing

These services created from service building blocks (SBBs) utilizing XML scripted service logic and a politic engine. Enabling the rapid design, development, customization and network-wide deployment of new services.

Genesys Telecommunications Laboratories and Veraz Networks have new options for connecting enhanced customer interaction and contact center applications existing network resources.

The integration of Genesys Voice services platform, with the Veraz ControlSwich softswitch allows service providers to connect IP-based customer interaction and contact center applications with existing Public Switched Telephone Network (PSTN) interfaces, using Primary Rate Interface (PRI) and SS7 ISDN User Part (ISUP). This pre-defined connection decreases costs and supports faster deployment of enhanced services for enterprises, including hosted call center applications. Using Veraz Control Switch for management and switching of traffic among different networks helps carriers provide the same set of applications through different networks.

ITU hold a workshop on next generation networks (NGN) together with the Internet Engineering Task Force (IETF) in Geneva.

Since May 2004 intense work has taken place in ITU, towards the development of standards that will define services, network and systems architecture in he next generation of IP enabled communication systems, or next generation networks (NGN). The objectives of the workshop are to report the progress of ITU's work on NGN and explore specific issues that impact both the ITU and the IETF in order to better understand the work underway in the two organizations and to identify areas where action can be taken to make further progress.

Analyzing of RESPIRE,
a novel approach to automatically blocking SYN flooding attacks

ANDRÁS KORN, JUDIT GYIMESI, DR. GÁBOR FEHÉR

Budapest University of Technology and Economics,
Department of Telecommunication and Mediainformatics, HSNLab

korn@chardonnay.math.bme.hu, gj309@hszk.bme.hu, gume@eik.bme.hu

Reviewed

Key words: attack, counting, flooding, syncookies

A few years ago, numerous major web sites were successfully brought down using an attack called SYN flooding. A number of methods for combating SYN floods have been proposed, many of which are widely deployed. In this paper, we describe a possible enhancement to some of these techniques; a way to automatically detect, isolate and filter SYN floods while conserving resources on the victim.

1. Introduction

The TCP SYN attack is made possible because establishing a TCP connection involves a so-called three-way handshake. The client starts the connection by sending a packet with the SYN flag set; it also specifies an Initial Sequence Number (ISN). The server replies to this with a packet that has both the SYN and the ACK flags set; it contains an acknowledgement for the ISN of the client and the ISN of the server. The connection is finalized when the client replies to this message with an ACK packet that acknowledges the ISN of the server.

In order for the server to be able to verify that the final ACK packet is indeed a reply to the SYN ACK, it has to compare the acknowledged sequence number with the ISN it gave the client; thus, it is necessary to establish a state when the SYN ACK packet is sent and to maintain it for some time: either until the final ACK arrives, or until it times out.

The attacker thus merely needs to send copious amounts of SYN packets (perhaps using spoofed source addresses). He or she ignores the SYNACK packets of the victim and never finalizes the connection. After a while, the finite connection backlog of the victim will be full and no further TCP connections to the attacked port will be possible.

2. Existing solutions

Some vendors (e.g. Cisco) offer routers that claim to offer protection against SYN floods. Some of these proxy the TCP handshake: they only send SYN packets to the protected server if they already received the final ACK. For the connection to work, the sequence numbers must be mangled on each subsequent packet of the session (because the router had to choose an initial sequence number for the connection, and the ISN of the server is bound to be different). These routers typically also have shorter timeouts on half-open connections and thus are indeed less vulnerable to SYN floods. It is important to note however that these approaches don't solve the actual problem, they merely increase the cost of a successful attack. It is still necessary to allocate finite resources (memory) for each connection. The only difference is that the attacker has to deplete the memory of the router, not the server.

Another suggested solution was to randomly drop SYN packets using a RED scheme [6]. Like shorter timeouts, RED also only makes the attack more expensive, but not impossible.

There are very general and thus somewhat heavyweight ways of dealing with flooding and congestion in general; one of these is described in [1].

An ingenious and widely deployed defense against SYN floods are TCP syncookies [3]. Syncookies work by sending a carefully crafted, cryptographically strong ISN back to the client in the SYN ACK packet, so that the ACKed sequence number in the final ACK packet is enough to validate the connection. No state is established and no memory used until the final ACK arrives. It is unfeasible for the client to guess a valid ACK sequence number and thereby spoof a connection without receiving a SYN ACK packet from the server first.

Detecting SYN floods is a different problem. One of the proposed solutions is described in [4] – while a stateless, "dumb" device does have its merits, the problem with this particular approach is that it can only help filter the flood if the device is located near the attacker. This means it would have to be deployed at every ISP worldwide in order to be useful. Failing that, the device can only detect that a flood is in progress but can't tell us who the perpetrator is.

3. Problems with syncookies

However, using syncookies has drawbacks. First of all, a connection established using syncookies cannot use large windows and can only use a fixed set of Maximum Segment Size (MSS) values. Second, syncookies take

time to compute: Bernstein suggests using the Rijndael algorithm to generate the ISN. Third, they magnify the effect of the SYN flood by responding with a flood of SYN ACK packets – possibly to unwitting third parties, if the flood uses forged source addresses. Thus, syncookies can actually make the situation worse by allowing "bounce flooding".

Therefore, even though syncookies ensure continued operation of a service even when under attack, it still makes sense to use a packet filter to prevent the offending SYN packets from reaching the server at all.

The approach presented in this paper is complementary to syncookies. The cookies can ensure that the service remains available while the *RESPIRE (Resource Efficient Synflood Protection for Internet Routers and End-systems)* mechanism reacts to the flood and blocks it; however, as shown below, reaction times are so short that syncookies are not strictly required.

4. How RESPIRE works

In contrast, our approach requires no additional data-gathering equipment to be deployed. Rather, it makes use of the data the victim itself must collect anyway in order to be able to provide TCP service.

The victim has a plethora of useful information we can use to determine whether we are under a SYN flooding attack; for example, we probably are if any of the following conditions are met:

– the number of incoming SYNs per second exceeds a threshold;
– a TCP backlog queue gets filled, so we have to start sending syncookies;
– the number of half-open connections exceeds a threshold;
– there is a disproportional difference between the number of SYN ACK packets sent out and ACK packets received.

RESPIRE as described here relies primarily on the last heuristic, but using a combination of all of the above is possible with minimal modifications.

Note that it would be possible to compare the number of arriving SYN packets to the number of inbound connection-finalizing ACK packets. However, in order to identify ACK packets that are indeed the last packet of a handshake, we need to track all TCP connections anyway, which involves analyzing the SYNACK packet and recording its ISN. Based on this information, we could reconstruct the SYN anyway, so processing the SYN packets separately seems redundant. However, in order for us to be able to rely on counting outbound SYNACK packets, the victim needs to be able to respond to a sufficient number of SYN flood packets with SYNACKs. Syncookies guarantee this ability, but if they can't be used for some reason, we must choose a backlog size that allows enough SYNACK packets to be sent for RESPIRE to identify the attackers before the backlog fills up. If this cannot be done, we can still count inbound SYN packets instead of outbound SYNACKs, but still need to process outbound SYNACK packets as well because we need their ISN.

So, to sum it up: it makes sense to count inbound SYNs instead of outbound SYNACKs if the protected server can use neither syncookies nor a sufficiently large backlog queue.

When we are under a SYN attack, the best we can do is to ignore the SYN packets of the attacker. The simplest way to accomplish this is to set up firewall rules that block SYN sent by the attacker; this means that our most important objective is identifying the address(es) the attacker uses. We could only do better than this by "pushing" the filtering towards the attacker along the network route his packets traverse towards us using *pushback* [5] or a similar mechanism.

Glossary

A.B.C.D/E
This is a shorthand notation for an IP subnet where the first E of 32 address bits identify the network, with the remaining bits identifying a node within that network. For example, the Budapest University of Technology and Economics uses the 152.66.0.0/16 network. "E" is commonly referred to as the "size" of the network. The smaller E is, the more nodes the network contains.

ACK
One of the flags used in TCP. Indicates that the packet contains an acknowledgement of previously received data.

cookie
Used to denote cryptographically generated data that is used in authentication.

DoS
Abbreviation of "Denial of Service". DoS attacks try to disable or sabotage a service.

SYN
One of the flags used in TCP. If set, the packet is referred to as a "SYN packet". The TCP handshake starts with the client sending a SYN packet.

SYNACK
The second packet of the three-way TCP handshake is commonly called a "SYNACK packet". It has both the SYN and the ACK flags set.

port
A two-byte endpoint identifier that is used by TCP and UDP to distinguish between network flows related to a single IP address.

RED
Random Early Drop. A congestion control mechanism that avoids congestion by dropping some packets before the network becomes congested.

spoofing
Forgery (of the source address of a packet).

sequence no.
Every data unit sent using TCP has a sequence number: basically the number of bytes transmitted so far plus a random offset determined at connection setup. The random offset makes connection forgery more difficult.

Historically, it used to be possible to forge just about any source address on a packet, so isolating the sources would not have been possible. By now, however, most networks have reverse path filters or are using other mechanisms to filter packets that are obvious fakes; therefore, an attacker can typically only forge addresses within one (or a handful of) class C network(s). We note that RESPIRE fails if the attacker can spoof any source address; in fact, it can exacerbate the situation by blocking legitimate clients in an attempt to block the attacker. Combining RESPIRE with spoof detectors like hop-count filtering [2] can significantly reduce this risk.

5. Anatomy of a SYN Flood

The typical attack scenario today is that the attacker has access to a number of computers compromised previously and now under his control – commonly referred to as "drones" – in several subnets around the world, and instructs all of them to launch an attack in concert, effectively mounting a distributed denial of service (DDoS) attack. To make filtering the packets more difficult, the drones use spoofed addresses, but every address is within the same netblock as the real address of the drone; otherwise, the edge router of the netblock would discard the packets.

Note that if the magnitude of the SYN flood is sufficient to flood the entire downlink of the victim, the attack is no longer a SYN attack but a generic bandwidth depletion attack that happens to use SYN packets; it is not our goal to deal with this scenario here.

Identifying the attacker

As mentioned earlier, we assume that during a SYN flood, the ratio of the number of outgoing SYN ACK packets to the number of incoming handshake-finishing ACK packets is going to be much larger than one. Note that most SYN ACK packets that go unacknowledged are sent to the attacker; thus, we can identify the attacker by finding the subnet with the most outgoing SYN ACKs per incoming ACKs.

A naive way of doing this would be to count the SYN ACK and ACK packets going to and coming from each class C subnet in a large table with 224 (16.7 million) entries. It is easy to see that this would be grossly inefficient; most of the counters would be zero, and most of those that are positive would only indicate benign behaviour. Finding the attacker would require looking at every entry in the table.

6. RESPIRE in detail

MULTOPS [7], the algorithm RESPIRE is loosely based on, addresses this problem by storing the counters in an efficient, dynamically expandable hierarchical data structure that exploits the hierarchical nature of IP space: a 256-ary tree is constructed to hold the counters.

The root of the tree contains two counters, initialized to zero, and 256 pointers, initialized to NULL. One of the counters, *Synack_Out*, counts the SYN ACK packets leaving the system. The other counter, *Ack_In*, counts the valid ACK packets (ones that finish TCP handshakes) entering the system.

After at least *Synack_Min* SYN ACK packets have been sent, the counters are consulted after each further *Synack_Num* SYN ACK packets are sent out. The tree structure makes this a relatively cheap operation to carry out. Because the number of tree levels is at most four, four divisions and comparisons and eight increments must be carried out per SYN ACK packet. For this reason, we recommend setting *Synack_Num* to one.

Sites with very large amounts of traffic can reduce the overhead by increasing *Synack_Num* at a small cost in flood detection speed and accuracy. Instead of deterministic sampling, stochastic methods can be used, or *Synack_Num* can be adjusted dynamically based on the amount of traffic received; however, these variations have no impact on the fundamental operation of the algorithm.

If the ratio of *Synack_Out to Ack_In* exceeds the value of the parameter R_{max} (a value of 1.5 or more is recommended), then in the last sampling period, the number of outgoing SYN ACK packets outnumbered the number of incoming ACK packets at least 1.5 to 1. This should not happen under normal circumstances, so we assume that we are under attack.

If we are under attack, we begin expanding the tree. For each *Synack_Num* subsequent outgoing SYN ACK or incoming ACK packet, we note the remote IP address A.B.C.D. If the root node pointer A is NULL, we allocate a new node with the same structure as the root node and link it to root→A. All SYN/ACK traffic associated with A.0.0.0/8 is from now on counted in both the root node and in the counters of root→A.

If root→A already exists, we check if

A→*Synack_Out* ≥ *Synack_Min*[L1] and if

$$\frac{\text{root}\rightarrow A\rightarrow \texttt{Synack\_Out}}{\text{root}\rightarrow A\rightarrow \texttt{Ack\_In}} > R_{\max}$$

If so, A.0.0.0/8 is probably one of the sources of the attack. We "zoom in" further by creating A→B if it doesn't already exist and so on until A→B→C exists.

The *Synack_Min* parameter can be different for each tree level. Decreasing the limit on the lower levels makes attack isolation faster but slightly less accurate. To compensate this, it would be possible to increase R_{max}. We plan to investigate such fine-tuning possibilities in a future paper.

If A→B→C exists, has at least *Synack_Min*[L3] SYN ACK packets associated with it and the ratio of its counters exceeds R_{max}, A.B.C is assumed to be an attacking subnet and is blocked, i.e. no further incoming SYN packets are accepted from A.B.C.0/24.

How this blocking is done is beyond the scope of this paper. The possibilities include, but are not limited to:

– Adding the filter to TCP stack of the OS kernel.
– Using the built-in packet filtering mechanisms of the underlying operating system, if any.
– Using a mechanism like *pushback* [5] to request filtering from an upstream router.

Naturally, these blocks should be temporary. After *Block_Timeout* minutes (15 recommended), they can be removed. This value should be chosen so that it is slightly longer than the typical flood is expected to be; it is unwise to set it too high, because having too many packet filtering rules puts a strain on the device that does the filtering. Also, continuing to block a subnet after the flood is over could result in blocking legitimate clients.

Once we find and block an attacking subnet, we remove its node from the tree; since we blocked it, we won't be receiving any further packets from it anyway, and we subtract its package counters from the counters in its parent nodes. This is done so that in the parent nodes the packet ratios more closely reflect the expected new distribution, where packets from the newly blocked subnet will no longer enter the system. This allows us to more accurately decide whether there still are other attacking subnets.

Once every *Prune_Interval* (2 seconds in our simulation), we check if the tree contains suspicious nodes (with counter ratios in excess of R_{max}). If so, we zero their counters. We remove all other nodes from the tree, except, obviously, the root node.

Only zeroing suspicious nodes instead of removing them allows us to more quickly identify them as attackers during the next *Prune_Interval*, because we don't have to wait for *Synack_Min* packets to accumulate in their parent nodes as the lower level node already exists.

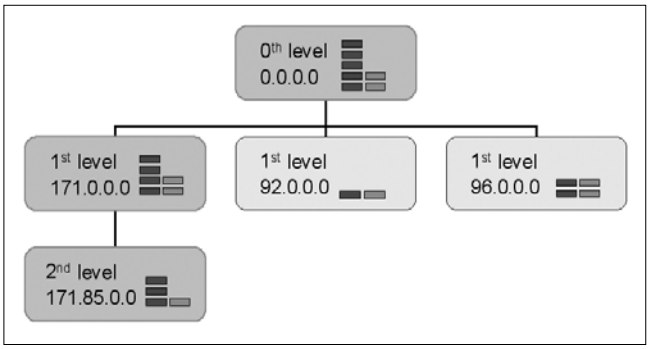
We zero the counters because we are only interested in ongoing flooding activity; we do not want past suspicious behaviour of a subnet to bias our future decisions. Unfortunately, this means that an attacker with a high number of distinct class C networks under her control can insinuate a "slow SYN flood" into the protected system; i.e. she can send less than *Synack_Min[L3]/Prune_Interval* packets per second from each subnet, so that none of them are identified as individual flood sources and blocked, but their cumulative effect on the service is detrimental nevertheless.

In this case, we can classify the nodes as "above suspicion", "slightly suspicious" and "definitely suspicious". Nodes are above suspicion if their counter ratio is smaller than an R_{legit} value (1.1 or even 1.05). These nodes we can remove from the tree at *Prune_Interval* boundaries. Slightly suspicious nodes have counter ratios between R_{legit} and R_{max}. We zero the counters of these nodes but don't remove them yet. A node is definitely suspicious if its counter ratio exceeds R_{max} but it didn't yet accumulate *Synack_Min* packets.

We don't even erase the counters of such nodes. It is reasonable to expect that after a while the nodes associated with the attackers will satisfy the criteria of filtering.

Figure 1. below shows an example RESPIRE tree. The two columns inside the nodes represent the relative amounts of SYN ACK and ACK packets respectively (but not an exact count). The nodes 92.0.0.0/8 and 96.0.0.0/8 sent approximately as many ACK packets as we sent them SYN ACKs, so they are probably benign. The darker nodes on the left, 171.0.0.0/8 and 171.85.0.0/16, are the suspicious ones. Note that the root node is also "suspicious"; this is what tells us that we are under attack.

Figure 1. Example of a RESPIRE tree



RESPIRE Memory Usage

In order to avoid memory exhaustion attacks against RESPIRE, the total number of tree nodes must be limited. One node occupies 2x64 + 256x64 bits: the two counters and the 256 pointers, assuming a 64-bit architecture. This adds up to 2064 bytes, or half that, about one kilobyte on more common 32-bit computers. The maximum number of nodes that could be created if no limit were enforced is 16777216 + 65536 + 256; thus, the total amount of memory used by the tree structure could increase to up to about 32.5 gigabytes (half this on 32-bit architectures), which is impractical to store and manage.

Our simulation showed that more than 150 nodes are seldom required even when the attackers command rather large address spaces. If we assume an unrealistic case where 200 different class C networks are used to flood the victim, and these all reside in different A blocks, only 600 nodes would have to be allocated, adding up to slightly more than one megabyte in size. Thus, limiting the amount of nodes to about 500 seems safe.

What to do when the limit is reached? If a new node is to be created beyond the 500th, find the least suspicious node under the root node and remove it and its children. If the root node only has one branch, continue the search on level A; obviously the single A node must have more than one branch, or we could not have 500 nodes in total. (More sophisticated methods could be used to find nodes to delete, but they would be more expensive.)

7. RESPIRE reaction time

In this section we will try to estimate the reaction time of the algorithm using mathematical methods. Let us first assume that we are dealing with a single attacking class C subnet. The calculations can be generalized to apply to more complicated cases, except "slow floods", which were discussed earlier.

In cases where multiple class C subnets are attacking, we can aggregate the times needed for each attacking subnet to be banned. This will be the worst case, because usually we should be able to ban several subnets in one go.

We assume that the attacking SYN packets arrive with an intensity of Ψ packets/sec. The legitimate traffic can be described as a λ parameter Poisson process, since legitimate users are independent. If we assume that burstiness is minimal, outbound SYNACK and inbound ACK packets likewise resemble Poisson processes, because the time needed for the computer to compose a SYNACK packet from an incoming SYN requests is approximately constant. This way, the Round Trip Time (RTT) does not cause a significant error in this approximation.

Even though the ACK packets arriving in a time interval are not necessarily the replies to the outgoing SYNACK packets of the same interval, the expected value of their number should be almost equal; with Poisson processes, we can expect a similar number of events in intervals of the same length, regardless of when the intervals start.

Let Δt be the time the attack begins after a *Prune_Interval* boundary. Two conditions are tested by the algorithm:

$$\frac{\lambda_l \cdot \Delta t + \psi_t \cdot \Delta t}{\lambda_l \cdot \Delta t} \geq R_{\max}$$

and

$$\lambda \cdot \Delta t + \psi_t \cdot \Delta t \geq synack\_min_{root}$$

The first inequality is independent of Δt; thus, if the above assumptions hold, we recognize an attack as soon as *Synack_Min* is exceeded. The condition of detection thus is:

$$\Psi_t \geq (R_{\max} - 1) \cdot \lambda_l$$

Thus the time needed for each tree level is:

$$\Delta t'_{level} = \frac{synack\_min_{level}}{\lambda_{level} + \psi}$$

Naturally, detection takes longer if, while counting, a *Prune_Interval* ends, because all counters are zeroed. Fortunately, counting can be resumed at the same tree depth we left off, because parent nodes are not erased. Using the indicator function *I{A}* which returns 1 if condition A is met and 0 otherwise, both cases (i.e. crossing an interval boundary or not) can be written in one equation. The condition tests whether the *Prune_Interval* the counting started in is different from the one it is supposed to end in. More than one boundary cannot be crossed; if the criteria of detection are met, we detect the flood in either one or two intervals.

$$\Delta t_{root} = I\left\{\left[\frac{\Delta t_t}{\Delta t_p}\right] < \left[\frac{\Delta t_t + \Delta t'_{root}}{\Delta t_p}\right]\right\} \cdot \left(\left[\frac{\Delta t_t}{\Delta t_p}\right] \cdot \Delta t_p - \Delta t_t\right) + \Delta t_{root},$$

Simplifications are possible by recognizing that if detection cannot be finished in the interval it started in, the time remaining until the next boundary is less than $\Delta t'_{level}$. In the next interval, we start counting again, and will take approximately $\Delta t'_{level}$ seconds to identify the attacker. The time spent at a tree level can thus be estimated by:

$$\Delta t_{level} \leq 2 \cdot \Delta t'_{level}$$

The probability of crossing an interval boundary is smaller if we set the interval length larger. A compromise must be found: the desire to only detect on-going attacks requires the interval to be small, whereas reaction times are better if intervals are longer.

When the root node indicates attack, we start building the tree. This procedure, along with the examination of the nodes, their manipulation, counting etc. do not cause a relevant time overhead, because packet transmission times are typically several orders of magnitude larger.

In order to better estimate the time spent at each tree level, we need to introduce a parameter "a" that indicates what fraction of the legitimate traffic originates from a given subnet as we move down the tree. Assuming that every subnet is responsible for an equal portion of the whole traffic would mean that packets from a specific class A subnet make up only about 1/256th of all the incoming SYN requests. For a class B subnet, the amount would be 1/256^2, for a class C subnet, 1/256^3. Naturally, this will not be true in practice. One reason is the distribution of IP addresses. At the A level, a significant portion of the address space is reserved for special purposes. The B and C address spaces are also unequally used, but the situation is not as bad as at level A. Thus an approximation where we use the parameter only at the level A, and the lower levels are taken to be homogeneous, appears to be acceptable. The value of "a" will vary, but probably be somewhere between 16 (local server used mostly within a relatively small country) and 128 (busy global server).

Total reaction time is the sum of the time spent at each of the four levels. These, if we needn't cross any *Prune_Interval* boundaries, and are using different *Synack_Min* values for each level, can be written as follows:

$$\Delta t'_{root} \geq \frac{synack\_min_{root}}{\lambda_l + \psi_t}$$

$$\Delta t'_A \geq \frac{synack\_min_A}{\frac{1}{a} \cdot \lambda_l + \psi_t}$$

$$\Delta t'_B \geq \frac{synack\_min_B}{\frac{1}{a \cdot 256} \cdot \lambda_l + \psi_t}$$

$$\Delta t'_C \geq \frac{synack\_min_C}{\frac{1}{a \cdot 256^2} \cdot \lambda_l + \psi_t}$$

If we also wish to account for the possibility of crossing interval boundaries, the equations become more complex:

$$\Delta t_{root} = I\left\{\left[\frac{\Delta t_t}{\Delta t_p}\right] < \left[\frac{\Delta t_t + \Delta t'_{root}}{\Delta t_p}\right]\right\} \cdot \left(\left[\frac{\Delta t_t}{\Delta t_p}\right] \cdot \Delta t_p - \Delta t_t\right) + \Delta t'_{root}$$

$$\Delta t_A = I\left\{\left[\frac{\Delta t_t + \Delta t_{root}}{\Delta t_p}\right] < \left[\frac{\Delta t_t + \Delta t_{root} + \Delta t'_A}{\Delta t_p}\right]\right\} \cdot \left(\left[\frac{\Delta t_t + \Delta t_{root}}{\Delta t_p}\right] \cdot \Delta t_p - \Delta t_t - \Delta t_{root}\right) + \Delta t'_A$$

$$\Delta t_B = I\left\{\left[\frac{\Delta t_t + \Delta t_{root} + \Delta t_A}{\Delta t_p}\right] < \left[\frac{\Delta t_t + \Delta t_{root} + \Delta t_A + \Delta t'_B}{\Delta t_p}\right]\right\} \cdot \left(\left[\frac{\Delta t_t + \Delta t_{root} + \Delta t_A}{\Delta t_p}\right] \cdot \Delta t_p - \Delta t_t - \Delta t_{root} - \Delta t_A\right) + \Delta t'_B$$

$$\Delta t_C = I\left\{\left[\frac{\Delta t_t + \Delta t_{root} + \Delta t_A + \Delta t_B}{\Delta t_p}\right] < \left[\frac{\Delta t_t + \Delta t_{root} + \Delta t_A + \Delta t'_C}{\Delta t_p}\right]\right\} \cdot \left(\left[\frac{\Delta t_t + \Delta t_{root} + \Delta t_A + \Delta t_B}{\Delta t_p}\right] \cdot \Delta t_p - \Delta t_t - \Delta t_{root} - \Delta t_A - \Delta t_B\right) + \Delta t'_C$$

The overall time needed obviously equals:

$$\Delta T = \Delta t_{root} + \Delta t_A + \Delta t_B + \Delta t_C$$

Let us consider the case where all legitimate SYN packets come from different class A subnets than the attacking ones. This gives us the worst case, since detection time depends only on the time needed to collect at least *Synack_Min* SYN packets – if the attack can be recognized at all. Under these circumstances we need not assume anything about the distribution of legitimate traffic among the subnets. We obtain:

$$\Delta t'_{root} = \frac{synack\_min_{root}}{\lambda_l + \psi_t}$$

$$\Delta t'_A = \frac{synack\_min_A}{\psi_t}$$

$$\Delta t'_B = \frac{synack\_min_B}{\psi_t}$$

$$\Delta t'_C = \frac{synack\_min_C}{\psi_t}$$

Reaction time when the algorithm crosses an interval boundary at each tree level can be written as:

$$\Delta T = 4 \cdot \Delta t_p - \Delta t_t + \frac{synack\_min_C}{\psi_t}$$

Please note that although reaction time seems to increase as *Prune_Interval* increases, its expected value actually decreases, as the earlier equations indicate. The reason is that when the interval is longer, the probability of reaching the interval boundary decreases.

Let us now give a rougher, but more compact approximation. According to our earlier observations, the time needed at each level is less, than double the time that would be required if no interval crossing took place. Total reaction time is thus smaller than the time we get by assuming that every level needs as much time as the level that needed the most time.

$$\Delta T \le 8 \cdot \max_{level}\left\{\frac{synack\_min_{level}}{\lambda_{level} + \psi}\right\}$$

Note that reaction time decreases as attack intensity increases. Reaction is practically immediate in the case of extreme floods, which cause the most damage.

This corroborates the results of the simulations; see below.

8. Simulation results

In order to analyze the performance of RESPIRE, we also ran a simulation [8]. For the sake of completeness, we sum up the results in this paper as well.

We simulated a busy SMTP server that has 62.8 active connections and 12.6 new connections per second on average. 300 simulated terminals were used to represent legitimate clients that randomized their IP before each connection. We also planted 8 attackers into the system who flooded the server with SYN packets. Using these attackers we modelled a distributed SYN attack. The state machine implemented in the attackers was different from the ones in the normal clients. The attackers use spoofed source addresses that are uniformly distributed across an entire subnet, the base address of which is a random value. The subnet mask is chosen randomly between 16 and 24; attacks that use entire /16 subnets should be very uncommon in practice, but we wanted to be generous with the attackers in order to put RESPIRE to a harder test. Each attacker performs one SYN attack of random length and intensity.

Table 1. (on the next page) summarizes some of the numerical results of the simulation.

Note that attacks #5 and #6 appear twice. This happened because these attacks lasted longer than the timeout for the firewall rules (15 minutes), so after the rules expired, these attacks had to be blocked again. “Unfiltered packets” shows the number of flood packets that passed RESPIRE by before the filtering rules took effect. The other columns should be pretty self-explanatory.

Let us now compare simulation results with our mathematical predictions.

Attacker #3 has a subnet of 4096 addresses (16 adjacent class C networks). This means we will see 16 suspicious class C nodes in the tree, all descendants of the same class B node. If the attacker chooses the source addresses of his packets uniformly, each of these nodes will account for 1/16th of the total flood intensity, that is, 3660.19 pps. Down to level B we can act as if we only had a single attacking class C:

Attack No.	First packet (s)	Subnet size	Packet rate (pps)	Unfiltered packets	Reaction time (s)
1	80,780	32768	41274	22191	0,629
2	239,046	512	91938	505	0,011
3	764,754	4096	58563	1920	0,045
4	890,803	512	82013	505	0,011
5	1229,573	16384	39586	6766	0,244
6	2039,08	8192	40932	3535	0,101
5*	2129,699	16384	39586	6767	0,165
6*	2939,157	8192	40932	3535	0,113
7	4060,253	32768	88895	13231	0,193
8	4729,277	512	31267	505	0,021

Table 1. Attacker activity

$$\Delta T_{root} + \Delta T_A + \Delta T_B \le 6 \cdot \max_{szint=root,A,B} \left\{ \frac{synack\_min_{szint}}{\psi_t} \right\} = ,01$$

At level C, we obtain:

$$\Delta T_C \le 2 \cdot \frac{synack\_min_C}{\psi_C} = 0,055$$

The total predicted time is thus 0.065 s which is a good estimate of the measured 0.045 s. The two methods produce comparable results.

Using the same methods, we can compute the predicted reaction times for all attacks: 0.635 s, 0.011 s, 0.65 s, 0.012 s, 0.338 s, 0.17 s, 0.032 s.

We needn't modify our upper estimate if several nonadjacent class C subnets start attacking at almost the same time. While this decreases the time needed for the root node to become suspicious, it doesn't influence the lower levels because the attackers reside in different class A nets. In our estimate, we used the time spent at the level where we spent longest, which certainly isn't the root node. Thus, the fact that the root node needs less time doesn't impact the rest of the calculations.

Naturally it can happen that the distribution of spoofed source addresses is non-uniform, which causes us to detect one attacking class C subnet before another. If some class C subnets use a substantially smaller attack intensity, total time taken can increase. Note however that in this case, the more damaging part of the flood has already been filtered, so it's acceptable to spend slightly more time blocking the rest.

9. Conclusion

In this paper, we introduced RESPIRE, one of several ways to combat SYN-floods. It appears to be a very lightweight solution that nevertheless filters SYN floods quickly and reliably. Additionally, it also reduces the amount of collateral damage a SYN flood can cause. Its memory requirements are very modest, rising above a few kilobytes only when under attack.

We suggest that RESPIRE be deployed alongside syncookies. A reference implementation for Linux is currently undergoing beta testing and will soon be released.

References

[1] Ratul Mahajan, Steven M. Bellovin, Sally Floyd, John Ioannidis, Vern Paxson, Scott Shenker, "Controlling High Bandwidth Aggregates in the Network" Computer Communications Review 32:3, July 2002, pp.62–73.

[2] Cheng Jin, Haining Wang, Kang G. Shin, "Hop-Count Filtering: An Effective Defense Against Spoofed DDoS Traffic", Proc. of the 10th ACM conference on Computer and communication security, 2003, pp.30–41.

[3] Daniel J. Bernstein, "SYN cookies", http://cr.yp.to/syncookies.html, 1997.

[4] Haining Wang, Danlu Zhang, Kang G. Shin, "Detecting SYN Flooding Attacks", Proceedings of IEEE InfoCom, 2002.

[5] John Ioannidis, Steven M. Bellovin, "Implementing Pushback: Router-Based Defense Against DDoS Attacks", Network and Distributed System Security Symposium, February 2002.

[6] Livio Ricciulli, Patrick Lincoln, and Pankaj Kakkar, "TCP SYN Flooding Defense", Comm. Networks and Dist. Systems Modeling and Simulation Conference (CNDS' 99), 1999.

[7] Thomer M. Gil, Massimiliano Poletto, "MULTOPS: a data-structure for bandwidth attack detection", Proc. of the 10th Usenix Security Symposium, August 2001.

[8] Gábor Fehér, András Korn, "RESPIRE – A novel approach to automatically blocking SYN flooding attacks", Proceedings of Eunice 2004, pp.181–187.

Implementation of the new european telecommunications regime

MARTIN CAVE

Warwick Business School, University of Warwick, UK
Martin.Cave@wbs.ac.uk

Key words: new regulatory framework, market analysis, Significant Market Power (SMP), obligations

Member states of the European Union, new and old, are now implementing the package of new regulatory arrangements for electronic communication services which was enacted in April 2002 and came into effect in July 2003. The processes involved are lengthy and require complex interactions between National Regulatory Agencies (NRAs) and the European Commission in Brussels. But enough experience has been gathered to make some preliminary judgements about the operation of the process. This note first gives a brief account of the new regime and then makes some comments about its operation in practice. These comments are based on an examination of the NRAs notifications to Brussels on individual markets, and on the author's experience in providing support to a number of NRA in implementing the procedures.

1. Outline of the new regime

The new regulatory framework for electronic communications was adopted in 2002. The main objectives of the new framework are to simplify the previous regimes, to apply them in a technologically neutral manner, and to encourage competition while guaranteeing user rights. Certainly, the previous regime has been streamlined, through a reduction from twenty key Community law measures to just five.[1]

At one level, the new régime is a major step down the transition path between regulated monopoly and normal competition, governed exclusively by generic competition law. As a result of the new regime, NRAs are no longer able to regulate the sector by issuing individual licences. Subject only to certain limited exceptions, Member States are required to establish a general authorisation regime. The conditions that may be imposed are heavily circumscribed. The new regime's provisions are applied across the range of 'electronic communications services', ignoring pre-convergence distinctions. It represents an ingenious attempt to corral the NRAs down the path of normalisation – allowing them, however, to proceed at their own speed (but within the uniform framework necessary for the internal market).

Since the end state is envisaged to be one governed by competition law, the European Commission proposes to move away from the rather arbitrary and piecemeal approach of the previous regulatory package towards something consistent with that law. However, competition law is to be applied (in certain markets) not only in a conventional responsive *ex post* fashion, but in a pre-emptive *ex ante* form. The new régime therefore relies on a special implementation of the standard competition triple of: market definition, identifying dominance, and formulating remedies to deal with (anticipated) competition law breaches. We examine these in turn.[2]

According to the underlying logic of the legislation, the Commission first establishes a list of markets where ex ante regulation is permissible, markets being defined according to normal competition law principles. These markets are then adapted and analysed by NRAs with the aim of identifying dominance (on a forward-looking basis). Where no dominance is found, ex ante obligations may not be imposed on any undertaking in the relevant market (ex post competition law would still apply). Where dominance is found, the choice of an appropriate remedy must be made from a specified list. The effect of the regime is to create a series of market-by-market 'sunset clauses' which reduce the level of ex ante regulation as the scope of effective competition expands.

1 *Directive 2002/19/EC on access to, and interconnection of, electronic communications networks and associated facilities [2002], OJ L 108/7 ("Access Directive");*
Directive 2002/20/EC on the authorisation of electronic communications networks and services [2002], OJ L 108/21 ("Authorisation Directive");
Directive 2002/21/EC on a common regulatory framework for electronic communications networks and services [2002], OJ L 108/33 ("Framework Directive");
Directive 2002/22/EC on universal service and users' rights relating to electronic communications networks and services [2002], OJ L 108/51 ("Universal Service Directive");
Decision No.676/2002/EC of 7 March 2002 on a regulatory framework for radio spectrum policy in the European Community ("Spectrum Decision").

2 *The first two of these processes are elaborated in respectively, Commission Recommendation 2003/311/EC of 11 February 2003 on relevant product and service markets within the electronic communications sector susceptible to ex ante regulation in accordance with Directive 2002/21/EC on a common regulatory framework for electronic communication networks and services. [2003]*
OJ C 114/45 ("Recommendation"); and Guidelines on Market Analysis and the assessment of SMP ("Guidelines On Market Analysis") [2002]
OJ C 165/6. Remedies are the subject of a Common Position on the approach to Appropriate remedies in the new regulatory framework [2004], ("the Remedies Paper") by the European Regulators Group, the college of NRAs created by the legislation to, inter alia, advise the Commission on the implementation of the procedures.

Market definition

In February 2003 the Commission issued a Recommendation on relevant markets, to identify those markets which, in the Commission's view, may warrant ex ante regulation. Unlike the previous regime, markets must be defined in accordance with the principles of competition law. NRAs may vary the markets subject to objection by the Commission. The Recommendation incorporates flexibility by allowing related 'technical services' to be aggregated within a market definition. Member States can also add or subtract markets, using specified (and quite complex) procedures.

The Recommendation identifies three cumulative criteria for identifying those markets which are suitable for ex ante regulation: high and non-transitory barriers to entry, the expected persistence of such barriers to entry over a relevant time period, making the prospect of effective competition unlikely, and the inability of competition law adequately to address the particular issue. The second of these is simply a projection of the first (albeit difficult to apply in practice). The logic of the régime thus rests heavily on the combined operation of the first and third criteria.

Dominance

Pursuant to Article 16 of the Framework Directive, the regulatory framework only permits the imposition of ex ante regulation where one or more undertakings is found to have *Significant Market Power* (SMP), which is identical to the standard definition of dominance determined and repeated by the European Court of Justice, ensuring in principle a major step forward towards the convergence of approaches under regulation and competition law.

Remedies

Under the Directives, NRAs have the power to impose obligations on firms found to enjoy SMP in a relevant market. Essentially, for wholesale markets the remedies are contained in Articles 9-13 of the Access Directive, while for retail markets the remedies are contained in Articles 17-19 of the Universal Service Obligations Directive. The wholesale remedies are, in ascending order of rigour: transparency, non-discrimination, separate accounting, mandatory access, and cost-oriented pricing. NRAs must act within a framework of duties set out in Article 8 of the Framework Directive and the measures they take must be proportionate to the policy objectives identified. This can be construed as meaning that the intervention is appropriate, no more than is necessary, and, by implication, satisfies a cost-benefit test, in the sense that the expected benefits from the intervention exceed the expected costs.

2. Experience

Over one hundred notifications of individual market have now been made to the Commission by NRAs, so it is possible to draw some preliminary conclusions about how things are going. I devide this assessment into *process* and outcome components, the latter divided between *market analysis* and *remedies*.

Process

The first point to make is that the regime imposes very heavy burdens on NRAs. In the UK, which has completed the process, an Ofcom official estimated that the reviews took about 60 person years of work. NRAs in smaller countries, which have the advantages of precedents, can reduce this vastly, but even in some of these the volume of analysis undertaken and length of notifications have been enormous. In my opinion, notifications have often contained unnecessarily exhaustive proofs of the obvious, and consideration should be given in future to streamlining the process.

The European Commission Task Force which receives the notifications (comprising officials from DG Comp and DF InfoSoc) also runs the risks of being swamped by the number of notifications (18 from each of 25 member states plus a few extras). The Commission has one month to accept a notification, with comments, or retain it for a further two months' study internally and by other NRAs through the Communications Council. So far, only a handful of notifications have gone to the second stage, and the Commission has required the withdrawal of only one market analysis – of the wholesale market for mobile access and call original market in Finland, where the NRA made a finding of dominance an the part of the largest operator, based largely on its market share in excess of 60%. The NRA must now resubmit the analysis.

Many NRAs have prenotification meetings with the Commission at which work in progress is discussed. These are unquestionably helpful and (combined with previous Commission comment) have almost certainly helped to reduce the number of notifications going to the second stage. NRAs reasonably infer that if an argument or piece of analysis submitted by another NRA has 'got through', the same approach will work for it if the circumstances are sufficiently similar.

Although the Commission's legal basis for approving an NRA's choice of remedies is much weaker than its basis for approving market definitions and analyses, its responses to notifications have also included comments on proposed remedies.

Market analysis

Despite the lengthy period taken over the analysis, the 'surprise' value of many of the notifications to date if very low. Broadly, we knew that competition was slow to develop in fixed networks which, tend to be dominated by the historic monopolist. This applies particularly to the smaller member states. The exceptions are national and, especially, international retail calls (especially by business customers, where the data permit such a distinction) and wholesale transit or conveyance on 'thick' routes. The same applies to leased lines at low speeds which are tied to the generally monopolised public switched telephone network.

An area of emerging interest, especially in relation to fixed markets, is whether competitive conditions in a member state are sufficiently uniform to justify a geographic market definition which covers the whole country, or whether separate regions should be distinguis-

hed, served by differing numbers of operators. For example, origination might be competitive on thick (inter-urban) routes but not on other routes. NRAs are reluctant at present to make such distinctions, but they may be necessary in the future.

Some comments on other markets are given below.

Fixed and mobile termination: in the Recommendation on relevant markets, these are defined as single operator markets, carrying the implication that each operator is a 100% monopolist. NRAs have so far accepted this approach, and it has led to the extension of the cost-based regulation currently found on fixed networks to termination on mobile networks too. As cost models are developed, several NRAs have proposed a 'glide path' of charges gradually reducing changes in a few years to a cost-based level. There is some question as to whether mobile networks of different sizes should have the same termination changes. In some cases, small, more vulnerable networks are allowed in the interim to set higher changes.

Mobile access and call origination: the Recommendation on relevant markets does not include retail markets for outgoing mobile services within the list of markets subject to ex ante regulation. However it does include the underlying wholesale market, despite the fact that, in the absence of national roaming, mobile virtual network operators (MVNOs) or wholesale airtime sales, there are no transactions on this market. Mobile operators do, however, supply themselves with such services, and this has formed a basis for discussion of whether there is single dominance on that market (as rejected by the Commission in the case of Finland – see above) or joint dominance exercised by two or more operators with similar market shares. Given the structure of mobile telephony in the EU, it is quite possible that one or more notifications of joint dominance may be made.

Wholesale international roaming: these are national markets (thus when a visitor from Hungary is in France, she cannot use a German networks to make and receive roamed calls), but with an international dimension: regulation in Hungary will, by definition, benefit visitors from other countries, not Hungarians. As a result, the European Regulators Group has put measures in place encouraging NRAs to co-operate with one another or conducting their market analyses. This process goes on simultaneously with a Commission competition investigation under Article 82 of the Treaty into the level of wholesale roaming charges set by two UK mobile operators.

Wholesale broadband access ('bitstream') and unbundled loops: these two markets are central to the competitive supply of DSL-based broadband services. While markets for local loops are likely to exhibit dominance, there is room for more debate about whether single (or possibly joint) dominance can be found in the market for bitstream in member states where there are developed cable networks and more than one operator which has installed broadband equipment in local exchanges.

Broadcast transmission services: NRAs have struggled to define and analyse this market, which might be taken to include some or all of a range of analogue and digital platforms relying on cable, DSL, satellite and terrestrial transmission. This market is likely to need review for future rounds of analysis.

Remedies

The ERG Remedies paper represents a laudable attempt by NRAs to provide guidance on remedies. But because they are not subject to notification of and approval by the Commission, it is harder to provide a synoptic view of the variety of remedies applied.

The challenge NRAs face in connection with choice of remedies is how best to use the flexibility available under the new, more narrowly defined anti-trust market and more focussed remedies. In my view, this is best achieved by adopting a zero-based approach – i.e. conjecturing how the market would operate without regulation. (This must in any case be done at the market analysis stage, where dominance is being tested for in a world without regulation.) Remedies to deal with problems can then be progressively added, and an estimate made of the incremental effect of each. The alternative is to start not from zero regulation, but from the *status quo,* and evaluate the effect of perturbations from that point. The problem here is that current remedies interact in various ways, and this approach may be too conservative in the sense that an NRA, not starting from a clean slate, might end up making no major change.

A second point, set out in the ERG remedies paper, is that it is extremely helpful if the NRA has a realistic understanding of how competition will develop over the period of the review and can gauge its interventions to help that process. This might involve opening up certain access points in the incumbents' networks, and withdrawing others where competitors have replicated the relevant assets. Unlike the case NRAs' market definitions and analysis, which can be evaluated at once on their own terms, the impact of remedies will be felt over a longer horizon. Nonetheless, an NRA can legitimately be criticised if it unthinkingly reproduces under the new regime all its current remedies.

3. Conclusion

The Commission is undertaking a review of the regime at the end of 2005, by which time its effect will be more evident. Tentatively, I would draw the following conclusions:

- the underlying logic of the new regime is sound, and fit for its long-terms deregulatory purpose;
- the process should be simplified except in the case of very difficult markets; this should go hand-in-hand with a reduction of the number of ex ante markets in the Recommendation; the result should be a speeding up of the process;
- so for the interactions between NRAs and the Commission have been effective and expeditious, but it remains to be seen if a faster flow of work can be dealt with;
- more thought can usefully be given to the design of remedies, with more systematic collection of effectiveness evidence, from member states.

Real-time charging in mobile environment

BÁLINT DÁVID ARY
ary.balint@isolation.hu
DR. SÁNDOR IMRE
imre@hit.bme.hu

Key words: content provision, UMTS, charging and billing, network structure

Charging the services offered by the packet based UMTS environment is much more complex, than it is in the circuit based GSM systems. The situation is even more complicated, as services can be easily developed using standard APIs and can be offered by third party providers. Moreover, pre-paid users need a real-time approach for charging and billing, which limits the admissible charging solutions. In our study we give a short survey of the motivations for developing the UMTS system, we summarize the legal and technical difficulties, and we introduce our new concept which should ease the majority of the problems and lighten the network overhead caused by the real-time charging solution presented in the corresponding technical reports.

1. Introduction

On the turn of the 20th and 21th century, the mobile telecommunication equipment went through a powerful development. After the early analogue systems the GSM (Global System for Mobile Communications) appeared, and the UMTS (Universal Mobile Telecommunication System) is being introduced nowadays in many countries in Europe. The evolution includes the increase of bandwidth, the fulfilment of the All-IP concept and the appearance of multimedia capable devices, thus different multimedia services like on-line streaming, video conferencing, Internet browsing and several packet based, on-demand services could be available with a next generation mobile phone. These changes were mostly called forth by the information society, which spends more and more money on communication. Even if these technical evolutions are not always urged by the subscriber demand, the new possibilities and functions are getting used widely.

Temporarily, the services accessible with mobile devices are offered by the network provider (who is the content and application provider as well), but with the continuous growth of the number of accessible functions and media it is predictable, that the network operator won't have enough time and/or energy to invent and offer new services, although it could lead to significant superiority in the market competition. With these conditions the supply of infrastructure and content should decouple, so content and applications should be served by 3rd party providers.

International telecommunication companies have made huge investments in UMTS, although a fully operating 3rd generation mobile network does not exist yet. The reason is that the changes are so significant, that the existing management systems are unable to handle these new demands. Not only the services should be developed but also the managing part should be revised, extended, and new features must be added, to realize the functions defined in the standards.

One of the main parts of the management system is charging. The return of the invested funds can only be hoped by new "killer-applications" and their proper accounting. The charging system of GSM networks was not designed to handle the bandwidth and the data/media types of UMTS services. A new charging concept should be developed, capable of handling the existence of 3rd party providers, and to support all kind of charging methods, like pre-paid, paynow, or post-paid mode.

The new architecture must be compatible with the existing GSM network charging architecture, and must operate real-time. Because the subscribers want to pay to only one provider (one-stop-shop concept), this accentuated provider has to maintain a financial relationship with the other providers and has to settle the invoices. The business model must be flexible, in order to support all combinations of content and provider relationships, hence the charging system must be accurate, convenient, transparent, and must be able to cooperate with other autonomous systems [6].

2. Business models

The amount of money paid after a service should be shared among the network operator and the content provider. But because the subscriber dislikes paying to more than one supplier for one service, the two providers should maintain some financial relationship, and settle the liabilities periodically after validation and identification. The content provider and the network operator have to agree on the parameters of the provided services (e.g.: required transport quality, parameters to be measured). Both providers must authenticate the user, and must know his or her financial status to decide, whether to accept or reject the service request.

Nevertheless, the rendition of the full account due to the user's personal rights is not viable.

Taking the business scope of 3rd generation networks, several business roles exist. The network operator provides access and transport services. The role of the content provider is to provide services, contents or applications that add value to transport services. These applications or contents can be produced by the content provider itself or purchased from other providers. The key function of the content aggregator is to package and offer services from one or several content providers. In third generation networks the key solution is flexibility [6]. A subscriber can belong to the content or the network provider, and both of them can charge the user.

Although several combinations of the roles and relationships are possible, the UMTS Forum outlined the most presumable business models [7]. A charging system must be prepared to deal with the different combination of business roles, and the charging systems of the providers must be compatible with each other.

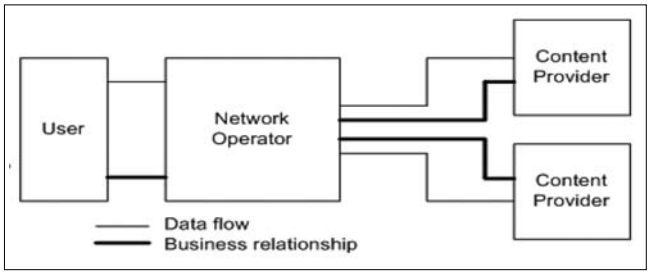


Figure 1. Network Operator Centric Business Model

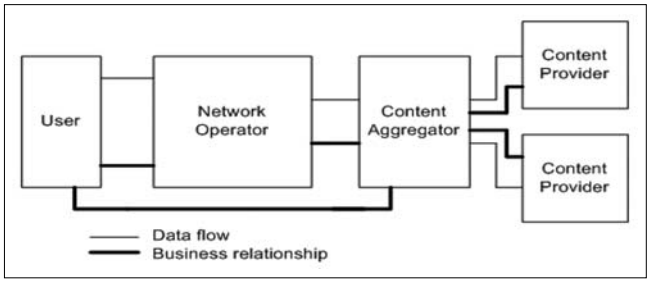


Figure 2. Content Aggregation Centric Business Model

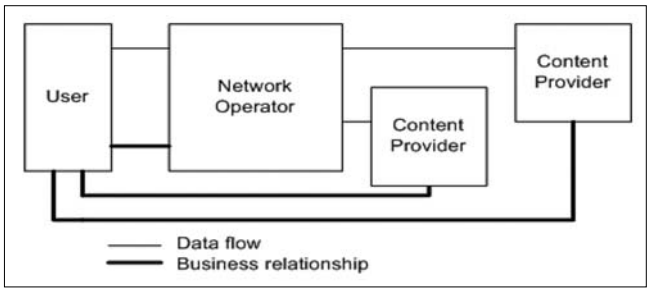


Figure 3. Content Provider Centric Business Model

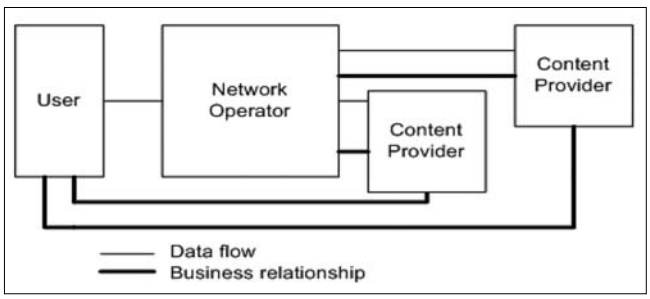


Figure 4. Hidden Network Operator Model

2.1. Network operator centric business model

In this model *(Figure 1.)* the network operator provides the content indirectly, charges the user and does the payoff to the 3rd parties. The subscriber can use the money on his/her infrastructure-account to pay for the content. In this way, content providing seems to the user like a value added network resource usage. The provider does not store the data and isn't responsible for it's content, therefore the Internet connection strongly determines the quality of the service (QoS). This model is the most convenient for the subscriber, although the operator has full control over the content providing.

2.2. Content Aggregation Centric Business Model

In the content aggregation centric business model *(Figure 2.)*, the content is accessed through a portal (which is not part of the mobile network). The service cost is split into two parts: the cost of the access to the aggregator (network resource usage) which is paid to the infrastructure provider, and the cost of the content accessed, which is paid to the aggregator. The fee of the content is defined by the content aggregator, who may be in connection with other content providers. In this way a chain of providers is involved in the transaction. To avoid the subscriber's chagrin, caused by the multiple payoffs, the content aggregator and the network operator should be in financial relationship and settle the bills among the 3rd party providers.

2.3. Content Provider Centric Business Model

The content provider centric business model *(Figure 3.)* is quite similar to the content aggregation centric business model, but the content provider plays the role of the content aggregator as well. Because of the huge number of 3rd party providers, the realization of the business relationship is much harder, than it was in the content aggregation centric business model. The main disadvantages of this solution are that the content/application providers must solve the accounting of services on their own, which can be more expensive than the service itself [10] and that the subscribers have to maintain an account with every content provider separately. This solution could lead to problems in case there are many providers. This model brings huge freedom to the services offered, but it means enormous administrative overhead as well.

2.4. Hidden Network Operator Model

In this model *(Figure 4.)*, the network operator stays unrevealed for the subscriber, as the content provider provides the mobile equipment, services and applications for the user and pays the necessary fees for the network operator. The model is suitable for companies with a very strong brand and with a tough customer base.

3. Technical challenges

In wired communication and in the circuit based GSM systems accounting was much easier. Because of the permanent and reserved bandwidth, the price of the service depends only on the length of the connection. The GPRS (General Packet Radio Service) and the UMTS are packet based, so the measurement of value and quality of the service are more complex.

In order to compute the quantity of the service, we should count the bits that have gone through the system. This method would require huge computing capabilities from the network elements (because of the high speed transport) and would impose a big overhead on the system (for N transmitted bits, the exact size can only be written in $\log_2 N$ bits). Counting of packets isn't the perfect solution either, inasmuch as the packet lengths in IP network vary in considerable range. Because of the defective quality of the Internet, a correct method should be aware of the lost and doubled packets. The additive cost because of these failures should not be charged to the subscriber.

In non-circuit switched systems the measurement of quality means problems as well, because in best-effort services a fixed reserved bandwidth is absent. Without a permanent connection a guarantee for capacity and delay can only be given with heavy signalling. In multimedia services the measurement of QoS is especially hard, because (f.e.) in case of video-streaming, the actual content affects the minimal requirements for the quality.

We showed that measuring quantity and quality is a quite complex task, which imposes notable overhead on the system. Furthermore, in a pre-paid environment, it must be done in real-time. In the present solutions, most providers solve it by combining data measuring with some easily measurable unit (time based accounting or constant bit rate accounting), or the measurement is done with greater scale (more kilobytes for instance) [9].

Another problem can be derived from the mobility. Every equipment has an IP address in the UMTS environment. If we use a fix IP address, and update the router tables in the network, the movement would be transparent for the charging mechanisms, but the router updates would cause overhead and signalling problems in the network. If the IP address changes continuously the network elements (which supply the charging information) must be informed respectively.

In the UMTS system, several media are accessible *(Table 1.)* [1]. Standards give possibilities to subscribers to possess separate accounts for all media available in the system. In case of 3rd party providers, if the accounting is done by the network provider, the operator should be aware of the exact method of charging and the measurable parameters of the service.

In 3G mobile networks it is possible for the users, to gain information about the price of the services, before the actual requisition. This supplementary service is ensured by the AoC (Advice of Charge) function. The mobile equipment retrieves charging-related information from the network, which contains the dependence of the price on service-time or service-data, or in case of an event based service (MMS for instance), on the entire, exact price of the service. This information is obviously sent by the same network element, which manages the charging and billing of the given service (the network operator or the 3rd party provider, depends on the business model used); nevertheless, if the information is sent by a 3rd party, the information should be validated and supervised [5].

4. Charging in UMTS networks

Besides the service and quality measurement the charging process also includes the settlement of invoices among the serving parties (subscriber, network and content provider). The price of network usage must also be settled between the network providers in case of roaming. This procedure is standardized, and uses the transfer account procedure (TAP) and a specific TAP format [1]. The construction of the bill presented to the user is also important; it must be simple and easily understandable [6]. The real money transaction between the parties (including the user) is usually obliged by contracts. These problems, solutions and mechanisms are beyond the scope of this article.

Standards define two different paying modes (prepaid and post-paid) and two charging methods: the offline and the online charging. The paying mode indicates the time, when the user has to pay for the service: afore or after the service require. The charging method indicates whether the subscriber's account is managed real-time (online) or not (offline). In offline charging, charging information is gathered after the requisition, and so, the subscriber account is debated after the service. Since this information is collected after the event/service, and sent trough a widespread network, real-time charging is not possible. The online charging mode assures that services are applied only if the subscriber has the necessary amount of money for them.

In offline mode the gateway (GGSN – Gateway GPRS Support Node) and the inner-nodes (SGSN – Serving GPRS Support Node) are sending charging in-

Table 1.

- speech,
- voice (real-time / streaming),
- video (real-time / streaming),
- data (download / upload / interactive content),
- messages (SMS / E-mail),
- data-flow (unspecified content),
- accessed web-pages, portals, etc.

Table 2.

- determinate data amount,
- determinate time-interval,
- the change of charging conditions,
- the change of QoS,
- the change of tariff,
- the change of position or cell,
- and the closure of voice, data or multimedia sessions

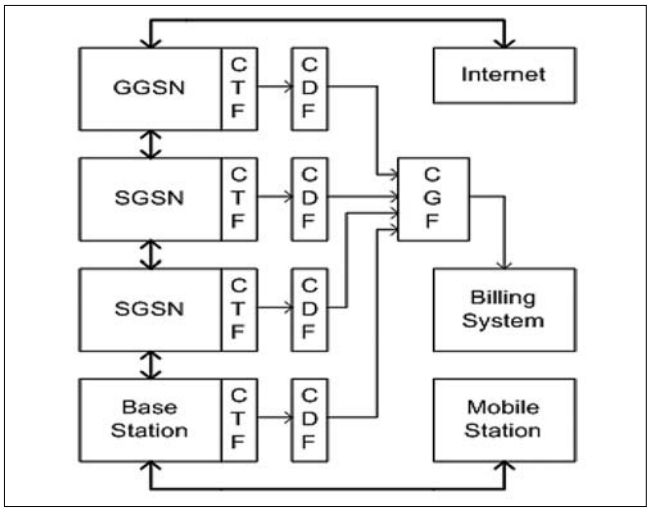


Figure 5. The offline charging architecture

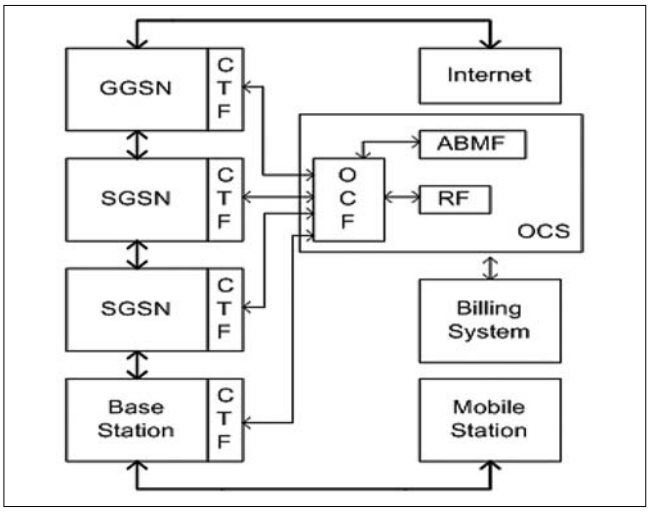


Figure 6. The online charging architecture

formation to the Billing System (BS). This charging information must be in standardized format, called Charging Data Record (CDR). The Charging Trigger Function (CTF) of the network elements generates charging events *(Table 2.)* based on the observation of network resource usage. The Charging Data Function (CDF) receives charging events from the Charging Trigger Function (CTF), and then uses the information contained in the charging events to construct CDRs. These records are sent to the Charging Gateway Function (CGF), which acts like a storage buffer, cleans, and preprocesses the CDRs. Finally, the CGF sends these processed CDRs to the Billing System *(Figure 5.)*. Because these charging records carry every information about the services required, the functionality of the CDRs extends beyond charging. With CDRs it's possible to analyze service-utilization, and gain statistical information about the services and content. By archiving the CDRs, the user-complaints can also be easily settled [1].

According to the standards [2], post-paid users can limit their account for a specific service; in light of this, a real-time charging method should be used for prepaid users and for post-paid users with credit limit. To ensure this, online charging should be applied.

In online mode *(Figure 6.)* the Online Charging System (OCS) is responsible for proper charging. The main task of this function is to realize real-time charging by continuously delegating certain amounts of credit to the serving network elements. These credits are deducted from the user's account. This method is called: unit reservation. If the service terminates before all credits are consumed, the network elements are retransfering the remaining credits to the OCS. To assure continuous service delivery, if the users do not terminate the service, a new amount of granted credit should be sent to the serving network element before the previous one runs out [3,4]. This unit-granting function is represented by the Online Charging Function (OCF) inside the OCS. The CTF generates charging events for the OCF as well, but this communication is bidirectional, as the OCF has to grant credits for the service. The OCS also includes the Account Balance Management Function (ABMF) and the Rating Function (RF). The Account Balance Management Function is the location of the subscriber's account balance within the OCS, and the Rating Function is used to determinate the value of the network resource usage and responsible for the

- rating of data volume (e.g. based on charging initiated by an access network entity),
- rating of session / connection time (e.g. based on charging initiated by a SIP application),
- and for rating of service events (e.g. based on charging of web content or MMS).

5. Mode-switching model

For a correct modelling it is obligatory to suit the related standards. The optimal model can be developed using the proper determination of the free parameters. Such variable parameters are the amount of data and/or time that triggers the CDR generation and the amount of granted credit during unit reservation. The smaller data/time amount we use, the more accurate the charging, and the larger the network overhead will be. It can be seen, that some trade-offs are necessary. Other variable parameters are the physical realization of the charging functions, inasmuch as these functions are not attached to hardware entities. The third variable parameter or method is the measurement of the services. The standards don't deal with the measuring methods, therefore for data transfer, the estimation of bandwidth or the exact bit count are possible solutions.

Since the unit reservation message should contain more or less the same information as the CDRs, we assume that they have the same size. In online charging, if the service reserves a large amount of credit from the user's account, access to additional, parallel resources could be denied, because there is no credit left on the account for another resource usage request; even if some service terminates afterwards, and the unused credits are returned to the users. In light of this, a more frequent unit reservation, with a smaller amount of cre-

dit should be applied. Because CDRs indicate the used services/data, this problem doesn't occur during offline charging. As follows, online charging causes bigger network overhead, than offline charging.

Our idea was not to glue the charging mode to the type of the payment (pre-paid, post-paid), but to dynamically switch between offline and online charging (if online charging is required) considering the user's account as well. Moreover, the overhead of the continuous unit reservation can also be reduced, by granting units only once. The quality of service should also be supervised, in order to charge services properly.

In our model, we assign a service specific limit to every service offered. If the user's account is above this limit, then charging is done in offline mode. If the subscriber's account drops below this limit, the online charging mechanism is applied (if required), and we grant all the consumable credit to the serving network element. In multi-task systems, it is possible to access more than one service at a time. In such cases, when the account drops below the limit, we shall delegate the credits to multiple network elements. A good solution is to distribute the account among the services with statistical methods, considering the money-consumption and properties of the services, and the behaviour of the user.

The UMTS services are based on a packet switched network, so we have to count with the packet-loss. The majority of these failures occur on the wireless part of the network but, of course (like on the regular Internet), some packet-loss or fault happens on the backbone as well. Statistical methods can be used to deal with these failures. Considering the quality of the operator's network, we can send more packets to the user, than it would be necessary with a perfect, flawless network, so the user presumably gets the proper amount of packets. It's practical to include a buffering mechanism between the wired and wireless part of the network, to cause the packets to be resent only from the base station in case of any failure, so the backbone isn't loaded with this traffic. The loss or fault occurring on the backbone can be solved with the error correction mechanism of the TCP – if necessary.

In order to measure the packet-loss and packet based QoS, the presence of trusted equipment is needed at the end of the connection. This could be the base station, or we can implement the protocol in the low level layer of the mobile phone. The main idea of this solution is that the element has to send some kind of information to the billing system, in order to inform it about the quality. The quality measurement of the data sequence is done with a sliding-window algorithm. After the arrival of a proper amount of packets, the delay (average, maximum, minimum and jitter), packet-loss, bandwidth and other QoS parameters can be calculated. The retransmission of the lost packets and signalling is done by higher protocols. The measurement of quality can be eliminated with the usage of pre-calculated statistical information for the network, but in this case the results won't mach the exact situation.

6. Analytical supplementation

Our model needs further refinement on order to determine the mode-switching limit, the handling of lost packets, and the measurement of QoS.

6.1. Mode-switching limit

Let us define a function called unit consumption speed

$$C(T), \tag{1}$$

having the measure of [*unit/sec*], which represents the consumed units in one second. The consumption rate depends on time to give the possibility to the operators to assign different prices to different time of the day and week for traffic shaping reasons. The consumed unit and money can be calculated from the consumption rate by means of the following equations

$$unit = C(T) \cdot \mathrm{t} \tag{2}$$

$$money = unit \cdot R(T), \tag{3}$$

where $R(T)$ represents the relation [2] between unit and money. The time-dependence of this function can be used to change the price of the units in case of inflation or discounts, or to apply different prices for different groups of users. Although the time dependence of the price can be divided into consumption speed and rating, it is not necessary, and it depends on the needs of the network operator.

Let T_c represent the time needed to query the user's proper account. The network elements are sending the CDRs usually in bigger time-intervals and the billing system debit the user's account periodically. T_c represents these intervals. With these notations and definitions the limit for mode-switch can be calculated. In ideal case it is

$$L = C(T) \cdot T_c. \tag{4}$$

If we own more units on our account than L, the charging is done offline with small network overhead; otherwise accounting is done online, with unit reservation. If we require more than one service at a time, the limit can be calculated by the sum of the limits of the services:

$$L = \sum L_i. \tag{5}$$

To reduce the network overhead, all credits below this threshold can be reserved. In case of multiple service demands, the units can be distributed to the serving network elements with the rate of the service's consumption speed. A re-sharing should be done every time a service ends, a new service started, or when an event based service occurs (SMS – for example). In order to ensure this, new functionality is required. The online charging function (OCF) should be able to force the network elements to retransfer the currently unused credits. After the transfer, the online charging function could re-share the credits among the services considering the new circumstances. When a fix consumption speed can not be assigned to the service (browsing or interactive content), the average consumption speed should be determined using various statistical models.

6.2. Propagation Delay

The events occurring in a distributed, wide network (signalling, queries) have propagation delay, which is not

constant in general. If we want to determine the mode-switching threshold properly, we have to consider the time needed the query the account (T_c) and to switch between modes (T_d), together with the variation of these values (T_{cj} and T_{dj}):

$$L = C(T)\cdot(T_c+T_{cj}+T_d+T_{dj}). \quad (6)$$

To ensure accurate charging, we should count with the maximum values of the jitters (T_{ci} and T_{di}). If we want to reduce the values of the mode-switching limits (in order to reduce the network overhead), we shall count with smaller values (with the expected value for example). In this case the possibility of users gaining more service than they paid for can be calculated from the distributions of the jitters. In case of re-sharing the control messages should be labelled with proper time-stamps to be able to charge the services gained during the retransfer and mode switching process.

The mode-switching thresholds can be calculated off-line for every service offered, and the system can use these pre-calculated values to switch between the charging modes. However, the actual limit can be dependant on the time of the day and on the user profile (discounts for group of users, statistical behaviour for interactive content).

6.3. Measurement of QoS

Performance can be defined using a sliding-window algorithm; always using the last N packets arrived to the user. With this method, the measured and experienced performance should be close to each other. Let t_j be the transmission starting time and a_j the arrival time of packet j. If the size of the sliding-window is N, the delay (average, minimum, maximum) can be calculated:

$$D_{average} = \sum(a_i - t_i) / N, \quad (7)$$

$$D_{min} = \min(a_i - t_i), \quad (8)$$

$$D_{max} = \max(a_i - t_i). \quad (9)$$

The jitter of the delay is the difference of the maximal and minimal delay:

$$D_{jitter} = D_{max} - D_{min}. \quad (10)$$

The packet-loss in case of N arrived, and M sent packets is:

$$Loss = N/M. \quad (11)$$

7. Conclusions and future plans

In our study, we enumerated the motivations for the appearance of 3rd party providers. We have showed the legal and technical issues of this new concept. Most of the technical problems come from the real-time nature and mobility in the packet based network. We also gave a small summary about the current state of the 3GPP standards, and finally, we gave a model to solve these problems. The model operates in such a way, that charging is made in the network offline, without a need for a real-time approach, to a large volume of users (who have more money on their account than the critical amount). This method invokes low CDR transfer, and low network overhead. Billing to critical users is more complicated, but also supported by 3GPP standards. With this idea the necessary network overhead can be decreased. Moreover, with a small function extension and statistical estimation, the overhead can be further reduced.

In the future, in order to develop the complete charging method, it is required to work out the exact method of measuring the data flow and the method to derive the quality of service from the IP based quality. For this, it is crucial to determine the statistic parameters of the services and users. The model is not complete unless the protocols and algorithms are fully developed.

References

[1] ETSI TS 122 115 V5.3.0 (2003-06). "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Service Aspects Charging and billing (3GPP TS 22.115, ver. 5.3.0, Rel. 5)." Technical report, 3GPP, 2003.

[2] ETSI TS 132 200 V5.7.0 (2004-06). "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Telecommunication management; Charging management; Charging principles (3GPP TS 32.200, ver. 5.7.0, Rel. 5)." Technical report, 3GPP, 2004.

[3] 3GPP TS 32.240 V6.0.0 (2004-09). "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; Charging management; Charging architecture and principles (Rel. 6)." Technical report, 3GPP, 2004.

[4] 3GPP TS 32.260 V2.0.0 (2004-12). "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; Charging management; IP Multimedia Subsystem (IMS) charging; (Rel. 6)." Technical report, 3GPP, 2004.

[5] ETSI TS 122 086 V5.0.0 (2002-06). "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Advice of Charge (AoC) supplementary services; Stage 1 (3GPP TS 22.086 version 5.0.0 Release 5)." Technical report, 3GPP, 2002

[6] UMTS Forum Report No.11. "Enabling UMTS 3rd Generation Services and Applications." Technical report, UMTS Forum, October 2000.

[7] UMTS Forum Report No.21. "Charging, Billing and Payment Views on 3G Business Models." Technical report, UMTS Forum, 2002.

[8] Maria Koutsopoulou, Alexandros Kaloxylos, Athanassia Alonistioti, Lazaros Merakos: "Charging, Accounting and Billing Management Schemes in Mobile Telecommunication Networks and the Internet." IEEE Communications Surveys, First Quarter 2004, 6(1), 2004, pp.50–58.

[9] Susana Schwartz: "Next-Gen Rating: It Will Be Only As Good as the Network." Billing World & OSS Today, February 2003, pp.16–22.

[10] John Cushnie: Charging and Billing for Future Mobile Internet Services, First Year PhD Research Report, September 2000.

SPEECH RESEARCH

Virtual speaker

Key words: facial animation, dynamic speech features, talking head, speechreading

Facial animation has progressed significantly over the past few years and a variety of algorithms and techniques now make it possible to create highly realistic characters. Based on the author's speechreading study and the development of 3D modelling, a Hungarian talking head has been created. Our general approach is to use both static and dynamic observations of natural speech to guide facial modelling. The evaluation of Hungarian consonants and vowels is presented for classifying visemes – the smallest perceptible visual units of the articulation process. A three level dominance model has been introduced that takes coarticulation into account. Each articulatory feature has been grouped to dominant, flexible or uncertain classes. The analysis of the standard deviation and the trajectory of the features served the evaluation process. Acoustic speech and articulation are linked with each other by a synchronising process. A filtering and smoothing algorithm has been developed for the adaptation either to the tempo of the synthesized or natural speech.

(In: 2005/1, pp.7–11.)

Design issues of a corpus-based speech synthesizer

Key words: synthesized speech, speech quality, sampling, corpus volume

The corpus-based approach is a new technique which has never been used in Hungary. It offers a more flexible and better quality synthesis. This article outlines the basic principles of this technique then a more detailed description follows of the development of a Hungarian corpus-based, object-related system being under development at the Speech Research Laboratory of the Budapest University of Technology and Economics. In the second part of the article statistical studies with weather forecasts are introduced then some considerations regarding the selection of announcers are presented. Finally some other design issues of corpus-based systems are addressed.

(In: 2005/1, pp.18–24.)

New protocol concept for wireless MIDI connections via Bluetooth

Key words: wireless, Bluetooth, MIDI (Musical Instrument Digital Interface)

This paper describes a new protocol concept for wireless MIDI connections via Bluetooth. For the practical appliance, the protocol design supports nearly arbitrary connection topology. We show the plans of a generic Bluetooth-based MIDI system and describe the main ideas of its data transmission protocol, calculate its latency and investigate its limits of usability, while suggesting a few possible extensions to this system to be further realized.

(–)

Introduction to Aspect-Oriented Programming

Key words: aspect-oriented programming (AOP), crosscutting concerns

Aspect-oriented programming is a fortunate extension to the wide-spread object-oriented paradigm. In this paper we present the most important concepts of AOP based on the most widely used AspectJ approach. The problem of crosscutting concerns is introduced, and the facilities provided by AOP are enumerated as possible solutions. The most popular implementations (HyperJ, Composition Filters) are also mentioned briefly.

(In: 2004/10, pp.8–12.)

SOFTWARE DEVELOPMENT RESULTS

Novel techniques for assessing resource requirements in packet-based networks

Key words: equivalent capacity, QoS, call admission control

The lack of quality of service (QoS) guarantees is the classic problem of packet switching networks. Despite the access technologies (e.g. DSL) providing sufficient transmission speed are already available, without such QoS guarantees the rapid spread of novel, value-added services can not be imagined. In this article a novel technique capable to approximate the minimum bandwidth that should be provided for an aggregated network traffic flow in order to maintain a predefined QoS level is introduced. This new method can form the basis of load control (e.g. call admission control) algorithms to be applied in future packet-based networks.

(In: 2004/9, pp.13–18.)

Self-adaptive Multimodal User Interfaces based on Interface-Device Binding

Key words: multi interface-device binding (MID-B), adaptive user interface, multimodal architecture

This paper introduces a mechanism that facilitates the dynamic shaping of human-machine interfaces in mobile environments. The necessity for self-adaptability of user interfaces for human-computer interaction (HCI) becomes increasingly important. Such adaptability provides the capacity to facilitate customised interaction depending on the system (and user) context based on automatic configuration of the user interface. The approach and work presented in this paper introduces a support architecture for self-adaptive user interfaces, based on distributed networked (interface) devices in a mobile communication scenario. The framework developed at the core of this work is called "Multi Interface-Devices Binding" (MID-B), it extends the concept of multimodality into the mobile environment, by allowing ad-hoc adaptation of available interface devices, (e.g. display, sound system, etc.). The here described structural design focuses on the definition of the basic architecture and of an extension to a service discovery protocol facilitating the dynamic real time binding of interface devices.

(–)